



TUGAS AKHIR - KI141502

RANCANG BANGUN APLIKASI PERANGKAT
BERGERAK LAYANAN PEMESANAN BARANG
**(Studi Kasus “Dinas Kebersihan dan Pertamanan
Kota Surabaya”)**

SHOFFI IZZA SABILLA
NRP 5112100132

Dosen Pembimbing I
Sarwosri, S.Kom., M.T.

Dosen Pembimbing II
Erma Suryani, S.T., M.T., Ph.D.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016



TUGAS AKHIR - KI141502

RANCANG BANGUN APLIKASI PERANGKAT BERGERAK
LAYANAN PEMESANAN BARANG
(Studi Kasus **“Dinas Kebersihan dan Pertamanan
Kota Surabaya”**)

SHOFFI IZZA SABILLA
NRP 5112100132

Dosen Pembimbing I
Sarwosri, S.Kom., M.T.

Dosen Pembimbing II
Erma Suryani, S.T., M.T., Ph.D.

JURUSAN TEKNIK INFORMATIKA
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2016

[Halaman ini sengaja dikosongkan]



UNDERGRADUATE THESES - KI141502

DESIGN IMPLEMENTATION OF APPLICATION
MOBILE FOR SERVICE BOOKING
**(Case Study “Dinas Kebersihan dan Pertamanan
Kota Surabaya”)**

SHOFFI IZZA SABILLA
NRP 5112100132

Supervisor I
Sarwosri, S.Kom., M.T.

Supervisor II
Erma Suryani, S.T., M.T., Ph.D.

DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2016

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

RANCANG BANGUN APLIKASI PERANGKAT BERGERAK LAYANAN PEMESANAN BARANG STUDI KASUS DINAS KEBERSIHAN DAN PERTAMANAN KOTA SURABAYA

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Rumpun Mata Kuliah Manajemen Informasi
Program Studi S-1 Jurusan Teknik Informatika
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh
SHOFFI IZZA SABILLA
NRP : 5112 100 132

Disetujui oleh Dosen Pembimbing Tugas Akhir

1. Sarwosri, S.Kom., M.T.
NIP: 19760809 200112 2 001 (Pembimbing 1)
2. Erma Suryani, S.T., M.T., Ph.D.
NIP: 19700427 200501 2 001 (Pembimbing 2)

SURABAYA
JULI, 2016

[Halaman ini sengaja dikosongkan]

RANCANG BANGUN APLIKASI PERANGKAT BERGERAK LAYANAN PEMESANAN BARANG STUDI KASUS DINAS KEBERSIHAN DAN PERTAMANAN KOTA SURABAYA

Nama Mahasiswa : Shoffi Izza Sabilla
NRP : 5112100132
Jurusan : Teknik Informatika FTIF-ITS
Dosen Pembimbing 1 : Sarwosri, S.Kom., M.T.
Dosen Pembimbing 2 : Erma Suryani, S.T., M.T., Ph.D.

Abstrak

Surabaya adalah kota terbesar kedua di Indonesia, sudah dapat dipastikan pula bahwa kota ini juga memiliki penduduk yang tidak sedikit. Untuk meningkatkan kesejahteraan masyarakatnya, pemerintah kota pastinya akan selalu mengupayakan proses pembangunan. Seperti yang terjadi di dinas kebersihan dan pertamanan kota Surabaya.

Secara keseluruhan dinas ini masih menggunakan cara manual dalam melakukan berbagai hal, misalnya pemesanan dan penyaluran barang ke tempat-tempat yang membutuhkan barang. Petugas lapangan harus mendatangi kantor dinas kebersihan dan pertamanan untuk melakukan pemesanan barang. Membutuhkan waktu yang lama untuk melakukan pemesanan barang, sehingga proses bisnis menjadi tidak efisien. Untuk menyelesaikan permasalahan tersebut dilakukan pembuatan aplikasi dengan perangkat bergerak. Aplikasi ini nantinya akan digunakan petugas lapangan untuk melakukan pemesanan barang sehingga tidak perlu datang ke kantor dinas kebersihan dan pertamanan kota Surabaya.

Pada tahap pengujian dilakukan dengan melakukan pemesanan barang dan manajemen data pemesanan secara runtut. Proses pengiriman dan penerimaan data untuk melakukan

pemesanan barang serta melihat status pemesanan juga diuji. Hasil dari pengujian yang telah dilakukan, aplikasi layanan pemesanan barang menjadi solusi terhadap permasalahan pemesanan barang yang kurang efektif. Hal ini berdasarkan dengan hasil kuisioner yang telah dilakukan dengan jawaban 60% sangat setuju dan 40% setuju.

Kata kunci:Perangkat Bergerak, Pemesanan, Android.

**DESIGN IMPLEMENTATION OF APPLICATION
MOBILE FOR SERVICE BOOKING
CASE STUDY DINAS KEBERSIHAN DAN
PERTAMANAN KOTA SURABAYA**

Student Name	: Shoffi Izza Sabilla
NRP	: 5112100132
Major	: Teknik Informatika FTIF-ITS
Supervisor 1	: Sarwosri, S.Kom., M.T.
Supervisor 2	: Erma Suryani, S.T., M.T., Ph.D.

Abstract

Surabaya is the second largest city in Indonesia, it is certain also that the city also has a population that is not little. To improve the welfare of its people, the city government will certainly always pursue the development process. As happened in the department of cleanliness and landscaping the city of Surabaya.

Overall the service is still using manual way of doing things, for example, ordering and delivery of goods to the places that need the goods. Field staff should go to offices of cleanliness and landscaping to make orders. Takes a long time to make orders, so that the business process becomes inefficient. To solve these problems made the creation of applications with mobile devices. This application will be used fieldworkers to an order of goods that do not need to come to the offices of cleanliness and landscaping city of Surabaya.

At this stage of the testing carried out by an order of goods and data management in a coherent ordering. The process of sending and receiving data to order goods and view order status was also tested. The results of the testing that has been done, the application service ordering goods into solutions to the

problems of ordering goods that are less effective. It is based on the results of questionnaires that have been carried out with the answer 60% strongly agree and 40% disagree.

Keywords: Mobile, Booking.

KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirabbil'alamin, segala puji bagi Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **“RANCANG BANGUN APLIKASI PERANGKAT BERGERAK LAYANAN PEMESANAN BARANG STUDI KASUS DINAS KEBERSIHAN DAN PERTAMANAN KOTA SURABAYA”**. Dengan pengerjaan tugas akhir ini, penulis bisa belajar lebih banyak untuk memperdalam dan meningkatkan apa yang telah didapatkan penulis selama menempuh perkuliahan di Teknik Informatika ITS. Dengan tugas akhir ini penulis juga dapat menghasilkan suatu implementasi dari apa yang telah penulis pelajari.

Selesaiannya tugas akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak. Sehingga pada kesempatan ini penulis mengucapkan syukur dan terima kasih kepada:

1. Allah SWT dan Nabi Muhammad SAW.
2. Keluarga penulis yang telah memberikan dukungan moral dan material serta do'a yang tak terhingga untuk penulis. Serta selalu memberikan semangat dan motivasi pada penulis dalam mengerjakan tugas akhir ini.
3. Ibu Sarwosri dan Ibu Erma selaku pembimbing yang telah membantu, membimbing, dan memotivasi penulis dalam menyelesaikan tugas akhir ini dengan sabar.
4. Bapak Dr. Darlis Herumurti, S.Kom., M.Kom. selaku Kepala Jurusan Teknik Informatika ITS.
5. Bapak Radityo Anggoro, S.Kom., M.Sc. selaku koordinator tugas akhir, dan segenap dosen Teknik Informatika yang telah memberikan ilmunya.
6. Bapak Chalid selaku Kepala Dinas Kebersihan dan Pertamanan kota Surabaya.

7. Teman-teman administrator laboratorium Algoritma dan Pemrograman (ALPRO), Mas Indra, Mas Yunus, Hanif, Demy, Ridho, Luffi, Yohana, Arvi, Aldi, Dimas, Rani, dan Sabila .
8. Nisrina, Satrio, Dwi, Hafi, dan Radhea Wicaksono yang menjadi pengingat dan penyemangat dalam proses pengerjaan tugas akhir.
9. Fadrian Merdianto dan Izzuddin yang membantu penulis dalam proses koding dan pengerjaan tugas akhir.
10. Teman-teman angkatan 2012 yang telah membantu, berbagi ilmu, menjaga kebersamaan, dan memberi motivasi kepada penulis, kakak-kakak angkatan 2011 dan 2010 serta adik-adik angkatan 2013 dan 2014 yang membuat penulis untuk selalu belajar.
11. Teman-teman HMTTC 2013/2014, khususnya Hublu yang telah memberi banyak kisah dan pengalaman berharga untuk penulis.
12. BPH Schematics ITS 2014 yang telah memberikan pengalaman penulis dalam pembuatan acara dengan skala nasional..
13. Serta semua pihak yang telah turut membantu penulis dalam menyelesaikan tugas akhir ini.

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depannya.

Surabaya, Juni 2016

DAFTAR ISI

LEMBAR PENGESAHAN	v
<i>Abstrak</i>	vii
<i>Abstract</i>	ix
KATA PENGANTAR.....	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xvii
DAFTAR TABEL	xxi
DAFTAR KODE SUMBER	xxiii
BAB I PENDAHULUAN	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	4
1.3 Batasan Masalah.....	4
1.4 Tujuan.....	5
1.5 Manfaat.....	5
1.6 Metodologi	5
1.7 Sistematika Penulisan Laporan Tugas Akhir.....	7
BAB II TINJAUAN PUSTAKA	11
2.1 Penelitian Terkait.....	11
2.2 Android.....	15
2.3 Java 15	
2.4 Android SDK.....	18
2.5 Kakas Kerja Android Studio	20
2.6 MySQL	20
2.7 Volley Library	23
2.8 JSON	23
2.9 JSON Web Token.....	24
BAB III ANALISIS DAN PERANCANGAN	27
3.1 Analisis.....	27
3.1.1 Domain Permasalahan	27
3.1.2 Deskripsi Umum.....	28
3.1.3 Aktor.....	29
3.2 Arsitektur Sistem	29
3.3 Kasus Penggunaan.....	31

3.3.1	Kasus Penggunaan Pemesanan Barang.....	33
3.3.2	Kasus Penggunaan Melihat Status Pemesanan	36
3.3.3	Kasus Penggunaan Melihat Riwayat Pemesanan...	37
3.3.4	Kasus Penggunaan Memperbarui Data Diri.....	39
3.3.5	Kasus Penggunaan Manajemen Pemesanan.....	40
3.3.6	Kasus Penggunaan Manajemen <i>User</i>	43
3.3.7	Kasus Penggunaan Melihat Semua Riwayat Pemesanan.....	45
3.4	Spesifikasi Kebutuhan Perangkat Lunak	46
3.4.1	Kebutuhan Fungsional Sistem	46
3.5	Perancangan Sistem	47
3.5.1	Perancangan Diagram Kelas	47
3.5.2	Perancangan Algoritma.....	50
3.5.3	Perancangan Basis Data	53
3.5.4	Perancangan Antarmuka	56
BAB IV IMPLEMENTASI		67
4.1	Implementasi Antarmuka.....	67
4.1.1	Antarmuka Aktivitas Login	67
4.1.2	Antarmuka Menu Utama.....	68
4.1.3	Antarmuka Pemesanan Barang	68
4.1.4	Antarmuka Lacak Pemesanan	72
4.1.5	Antarmuka Status Pemesanan	72
4.1.6	Antarmuka Riwayat Pemesanan	74
4.1.1	Antarmuka Informasi Kantor	74
4.1.1	Antarmuka Menu Profil Diri.....	76
4.1.2	Antarmuka Halaman Login Web	76
4.1.3	Antarmuka Halaman Status Pemesanan.....	77
4.1.4	Antarmuka Halaman Riwayat Pemesanan.....	80
4.1.5	Antarmuka Halaman Manajemen User.....	80
4.2	Implementasi Fitur.....	82
4.2.1	Fitur Pemesanan Barang	82
4.2.2	Fitur Melihat Status Pemesanan.....	84
4.2.3	Fitur Melihat Riwayat Pemesanan	84
4.2.4	Fitur Memperbarui Data User	86
4.2.5	Fitur Manajemen Pemesanan	87

4.2.6	Fitur Melihat Semua Riwayat Pemesanan	88
4.2.7	Fitur Manajemen User	89
BAB V UJI COBA DAN EVALUASI		91
5.1	Lingkungan Uji Coba	91
5.2	Dataset Uji Coba.....	92
5.3	Skenario Pengujian.....	92
5.3.1	Pengujian Fungsionalitas	93
5.3.2	Pengujian Performa Sistem.....	110
5.3.3	Pengujian Kepada Pengguna	111
5.4	Evaluasi Pengujian	113
BAB VI KESIMPULAN DAN SARAN.....		115
6.1	Kesimpulan.....	115
6.2	Saran.....	115
DAFTAR PUSTAKA.....		117
LAMPIRAN A KODE SUMBER.....		119
LAMPIRAN B DATA PENDUKUNG.....		135
BIODATA PENULIS.....		137

[Halaman ini sengaja dikosongkan]

DAFTAR GAMBAR

Gambar 2.1 Alur Pemesanan Barang	11
Gambar 2.2 Antarmuka Aplikasi Zahir POS.....	13
Gambar 2.3 Antarmuka Penjualan	14
Gambar 2.4 Antarmuka Pembayaran	14
Gambar 3.1 Arsitektur Sistem	30
Gambar 3.2 Kasus Penggunaan.....	32
Gambar 3.3 Diagram Aktivitas Pemesanan Barang	35
Gambar 3.4 Diagram Aktivitas Melihat Status pemesanan....	37
Gambar 3.5 Diagram Aktivitas Melihat Riwayat Pemesanan	38
Gambar 3.6 Diagram Aktivitas Memperbarui Data Diri.....	40
Gambar 3.7 Diagram Aktivitas Manajemen Pemesanan.....	42
Gambar 3.8 Diagram Aktivitas Manajemen <i>User</i>	44
Gambar 3.9 Diagram Aktivitas Melihat Semua Riwayat Pemesanan	46
Gambar 3.10 Diagram Kelas Aplikasi Perangkat Bergerak ...	49
Gambar 3.11 Diagram Alir Algoritma Pemesanan Barang....	51
Gambar 3.12 Diagram Alir Algoritma Status Pemesanan.....	52
Gambar 3.13 <i>Conceptual Data Model</i>	54
Gambar 3.14 <i>Physical Data Model</i>	55
Gambar 3.15 Perancangan Antarmuka Login	56
Gambar 3.16 Perancangan Antarmuka Menu Utama.....	57
Gambar 3.17 Perancangan Antarmuka Pesan Barang.....	58
Gambar 3.18 Perancangan Antarmuka Melihat Status Pemesanan	59
Gambar 3.19 Perancangan Antarmuka Melihat Riwayat Pemesanan.....	60
Gambar 3.20 Perancangan Antarmuka Informasi Kantor	61
Gambar 3.21 Perancangan Antarmuka Profil Diri	62
Gambar 3.22 Perancangan Antarmuka Login Web.....	63
Gambar 3.23 Perancangan Antarmuka Manajemen Pemesanan	64
Gambar 3.24 Perancangan Antarmuka Seluruh Riwayat Pemesanan.....	64
Gambar 3.25 Perancangan Antarmuka Manajemen Data User	65

Gambar 4.1 Antarmuka Login.....	67
Gambar 4.2 Antarmuka Menu Utama	68
Gambar 4.3 Antarmuka Pilih Kategori Barang	69
Gambar 4.4 Antarmuka Pemesanan Barang.....	69
Gambar 4.5 Antarmuka Memasukkan Jumlah Pemesanan	70
Gambar 4.6 Antarmuka Keranjang Belanja	71
Gambar 4.7 Antarmuka Konfirmasi Pemesanan	71
Gambar 4.8 Antarmuka Lacak Pemesanan.....	72
Gambar 4.9 Antarmuka Melihat Status Pemesanan	73
Gambar 4.10 Antarmuka Detail Status Pemesanan.....	73
Gambar 4.11 Antarmuka Melihat Riwayat Pemesanan.....	74
Gambar 4.12 Antarmuka Detail Riwayat Pemesanan	75
Gambar 4.13 Antarmuka Informasi Kantor.....	75
Gambar 4.14 Antarmuka Profil Diri.....	76
Gambar 4.15 Antarmuka Halaman Login Web.....	77
Gambar 4.16 Antarmuka Status Pemesanan.....	77
Gambar 4.17 Antarmuka Edit Status Pemesanan	78
Gambar 4.18 Antarmuka Detail Riwayat Pemesanan	79
Gambar 4.19 Antarmuka Halaman Rekap Pemesanan.....	80
Gambar 4.20 Antarmuka Halaman Manajemen User.....	81
Gambar 4.21 Antarmuka Halaman Tambah User	81
Gambar 4.22 Antarmuka Edit User	82
Gambar 5.1 Pengujian Memilih Kategori Barang	94
Gambar 5.2 Pengujian Memilih Barang.....	95
Gambar 5.3 Pengujian Memasukkan Jumlah Barang.....	95
Gambar 5.4 Pengujian Tombol Masukkan Keranjang Belanja	96
Gambar 5.5 Pengujian Tombol Keranjang Belanja.....	96
Gambar 5.6 Pengujian Hapus Barang.....	97
Gambar 5.7 Pengujian Tombol Pesan	97
Gambar 5.8 Pengujian Melihat Status Barang.....	99
Gambar 5.9 Pengujian Memilih Pemesanan.....	99
Gambar 5.10 Pengujian Melihat Riwayat Pemesanan.....	101
Gambar 5.11 Pengujian Memilih Riwayat Pemesanan	101
Gambar 5.12 Pengujian Memperbarui Data Diri.....	103
Gambar 5.13 Pengujian Melihat Status Pemesanan	104

Gambar 5.14 Pengujian Lihat Status Pemesanan	105
Gambar 5.15 Pengujian Edit Status Pemesanan.....	106
Gambar 5.16 Pengujian Melihat Semua Riwayat Pemesanan	107
Gambar 5.17 Pengujian Menambah User.....	109
Gambar 5.18 Pengujian Edit User	110

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 2.1 Perbandingan Aplikasi Sejenis	12
Tabel 3.1 Kode Kasus Penggunaan	33
Tabel 3.2 Spesifikasi Pemesanan Barang	33
Tabel 3.3 Spesifikasi Melihat Status Pemesanan	36
Tabel 3.4 Spesifikasi Melihat Riwayat Pemesanan	37
Tabel 3.5 Spesifikasi Memperbarui Data Diri	39
Tabel 3.6 Spesifikasi Manajemen Pemesanan	41
Tabel 3.7 Spesifikasi Manajemen <i>User</i>	43
Tabel 3.8 Spesifikasi Melihat Semua Riwayat Pemesanan	45
Tabel 3.9 Kebutuhan Fungsional Sistem	47
Tabel 3.10 Perancangan Aplikasi Perangkat Bergerak	56
Tabel 3.11 Perancangan Antarmuka Aplikasi Web	62
Tabel 5.1 Lingkungan Uji Coba	91
Tabel 5.2 Pengujian Fitur Pemesanan Barang	93
Tabel 5.3 Pengujian Melihat Status Pemesanan	98
Tabel 5.4 Pengujian Fitur Melihat Riwayat Pemesanan	100
Tabel 5.5 Pengujian Fitur Memperbarui Data Diri	102
Tabel 5.6 Pengujian Fitur Melihat Status Pemesanan	104
Tabel 5.7 Pengujian Fitur Mengedit Status Pemesanan	105
Tabel 5.8 Pengujian Fitur Melihat Semua Riwayat Pemesanan	107
Tabel 5.9 Pengujian Fitur Menambah User	108
Tabel 5.10 Pengujian Fitur Mengedit User	109
Tabel 5.11 Pengujian Waktu Komunikasi <i>Client-Server</i>	111
Tabel 5.12 Daftar Pengguna Uji Coba	112
Tabel 5.13 Hasil Uji Coba	112
Tabel B.9.1 Contoh Data Barang di Gudang	135
Tabel B.9.2 Data Pegawai Dinas Kebersihan dan Pertamanan	135
Tabel B.9.3 Data Kategori Barang	136

[Halaman ini sengaja dikosongkan]

DAFTAR KODE SUMBER

Kode Sumber 2.1 Format JSON.....	24
Kode Sumber 2.2 Header JSON Web Token	25
Kode Sumber 2.3 Claim JSON Web Token.....	25
Kode Sumber 4.1 Fitur Pemesanan Barang.....	84
Kode Sumber 4.2 Fitur Melihat Status Pemesanan	84
Kode Sumber 4.3 Fitur Melihat Riwayat Pemesanan	85
Kode Sumber 4.4 Fitur Memperbarui Data User	87
Kode Sumber 4.5 Fitur Manajemen Status Pemesanan.....	88
Kode Sumber 4.6 Fitur Melihat Semua Riwayat Pemesanan.....	89
Kode Sumber 4.7 Fitur Manajemen User.....	90
Kode Sumber A.8.1 Menangani Session.....	121
Kode Sumber A.8.2 Menampilkan Pilihan Barang	124
Kode Sumber A.8.3 Menampilkan Keranjang Belanja	127
Kode Sumber A.8.4 Menampilkan Status Barang.....	130
Kode Sumber A.8.5 Menampilkan Riwayat Pemesanan.....	133

[Halaman ini sengaja dikosongkan]

BAB I

PENDAHULUAN

Pada bab ini akan dipaparkan mengenai garis besar tugas akhir yang meliputi latar belakang, tujuan, rumusan dan batasan permasalahan, metodologi pembuatan tugas akhir, dan sistematika penulisan.

1.1 Latar Belakang

Surabaya adalah kota terbesar di Jawa Timur, sudah pasti akan selalu mengupayakan proses pembangunan untuk meningkatkan kesejahteraan masyarakatnya. Proses yang dilaksanakan tidak semata mengejar laju pertumbuhan ekonomi yang tinggi, tetapi harus diimbangi dengan pertumbuhan dibidang lain. Pertumbuhan dibidang lain dibutuhkan agar dapat mewujudkan kesejahteraan masyarakat secara optimal yaitu melalui rencana pembangunan dan pemeliharaan kota yang berkelanjutan. Dalam pembangunan kota Surabaya, pemerintah kota Surabaya dibantu oleh beberapa dinas pemerintah yang memiliki ranah kerja yang berbeda-beda. Terdapat 27 dinas di kota Surabaya salah satunya adalah dinas kebersihan dan pertamanan. Surabaya merupakan salah satu kota metropolitan yang ada di Indonesia. Banyak tempat-tempat pembuangan sampah yang salah penempatannya atau sudah tidak memadai lagi. Selain itu, ditambah masyarakat yang kurang memahami lingkungan lebih memilih untuk membuang sampah tidak pada tempatnya. Hal ini berdampak pada lingkungan menjadi tidak sehat dan keindahan tatanan kota Surabaya menjadi tidak sedap dipandang. Hal ini terjadi karena adanya beberapa faktor yaitu kurangnya tingkat kesadaran untuk menjaga lingkungan, kurangnya partisipasi masyarakat, sistem pengawasan, kurangnya koordinasi dan pemberdayaan dinas kota Surabaya dalam menyalurkan barang yang dibutuhkan di lingkungan tersebut. Dapat dilihat di jalan-jalan besar yang merupakan pusat keramaian kota, beberapa tempat sangat sulit untuk mencari

tempat sampah. Tak sedikit pula tempat sampah yang ada sudah tidak layak pakai sehingga harus di ganti dengan yang baru.

Selain masalah sampah di kota Surabaya, terdapat masalah lain yang dapat mengganggu keindahan tatanan kota Surabaya. Penerangan jalan umum adalah sesuatu yang penting untuk membantu kesejahteraan masyarakat. Jika tidak ada penerangan jalan umum bisa dipastikan jalan-jalan di kota Surabaya akan sangat rawan sekali dengan kecelakaan lalu lintas dan kejahatan lainnya. Sesuai dengan fungsi pokok dinas kebersihan dan pertamanan kota Surabaya yaitu, pertama adalah adalah pelaksanaan fasilitasi dalam pengelolaan kebersihan, pertamanan, penerangan jalan umum dan dekorasi kota, pemakaman, pembibitan, pengolahan sampah, air limbah rumah tangga dan lumpur tinja. Kedua, pelaksanaan pengadaan atau pembangunan aset yang akan digunakan dalam rangka penyelenggaraan tugas pokok. Ketiga, pelaksanaan pemeliharaan barang milik daerah yang digunakan dalam rangka penyelenggaraan tugas pokok. Berkaitan dengan sangat pentingnya penerangan jalan umum maka tindakan perbaikan yang dilakukan juga harus cepat. Penanganan akan dilakukan dengan cepat apabila ketersediaan barang dan lokasi yang dekat dengan kantor. Apabila lokasinya jauh maka memerlukan waktu yang panjang untuk meminta perbaikan karena jika ingin mengambil barang diharuskan untuk ke kantor terlebih dahulu. Tidak berhenti disitu di kantor tidak langsung menerima barangnya akan tetapi akan di cek terlebih dahulu ketersediaanya di beberapa gudang yang ada di kantor.

Begitu juga pada persoalan pertamanan di kota Surabaya, setelah mendapatkan beberapa kali adipura pemerintah kota Surabaya semakin menggecarkan perbaikan taman-taman kota yang ada. Dinas kebersihan dan pertamanan juga bertanggung jawab atas keindahan taman kota. Mulai dari penyaluran tanaman-tanaman, pupuk, pot, dan dekorasi taman guna menunjang keindahan taman kota.

Dapat dilihat sebagian besar fungsi pokok pada dinas kebersihan dan pertamanan adalah mengenai barang-barang yang

akan di salurkan ke beberapa tempat yang ada di kota Surabaya guna membantu keberhasilan kota Surabaya dalam bidang kebersihan dan pertamanan kota. Lama nya proses penyaluran barang akan mempengaruhi keberhasilan kota Surabaya dalam hal kebersihan dan keindahan kota. Untuk menunjang fungsi utama pada dinas kebersihan dan pertamanan kota Surabaya diperlukan sebuah perangkat lunak khususnya untuk pemesanan barang sehingga semua fungsi dapat berjalan dan termonitoring dengan baik. Saat ini dinas kebersihan dan pertamanan telah memiliki sebuah perangkat lunak yang hanya dapat digunakan oleh kepala pergudangan dinas kebersihan dan pertamanan. Perangkat lunak yang sudah ada ini hanya dapat melakukan proses pengadaan barang yang datang, pencatatan pemesanan dan menampilkan beberapa laporan.

Untuk pemesanan barang masih menggunakan cara manual yaitu petugas harus mendatangi kantor dinas kebersihan dan pertamanan untuk melakukan permohonan pemesanan barang. Pegawai gudang akan mencatat data pemesan dan barang yang ingin dipesan. Pegawai akan memeriksa terlebih dahulu apakah barang yang diminta berada digudang atau tidak. Selain itu pegawai memeriksa barang yang tersedia apakah mencukupi dengan jumlah yang diminta petugas. Jika barang yang diminta tersedia, petugas akan mendapatkan surat bukti pemesanan barang. Untuk mendapatkan barang yang diminta, petugas harus menunggu setidaknya satu minggu setelah menerima surat bukti pemesanan barang. Membutuhkan waktu yang lama untuk melakukan pemesanan barang sehingga proses bisnis menjadi tidak efisien.

Perkembangan teknologi bergerak (*mobile*) yang begitu pesat dalam beberapa tahun belakangan menciptakan revolusi besar pada perangkat ponsel pintar. Kebiasaan manusia dalam menggunakan komputer pun kini sudah mulai beralih ke arah perangkat bergerak. Kemudahan serta kecanggihan perangkat pintar baik itu ponsel maupun tablet membuat para penggunanya lebih sering menggunakan perangkat ini ketimbang komputer

desktop maupun *notebook*. Beragam aplikasi pada perangkat bergerak memudahkan dan mempercepat pekerjaan. Ditambah dengan adanya internet yang memiliki kemampuan untuk menghubungkan perangkat satu dengan yang lainnya. Meledaknya pengguna internet juga beriringan dengan meledaknya perangkat bergerak (*mobile*).

Dari permasalahan diatas, dapat ditangani dengan solusi penggunaan *smartphone* untuk mempermudah petugas maupun pegawai dinas kebersihan dan pertamanan dalam melakukan pemesanan barang. Sehingga diharapkan dengan diterapkannya solusi ini, dinas kebersihan dan pertamanan kota Surabaya dapat memecahkan masalah yang ada sebelumnya dalam melakukan pemesanan barang dengan baik dan efisien.

1.2 Rumusan Masalah

Rumusan masalah pada pembuatan tugas akhir ini adalah sebagai berikut:

1. Bagaimana membangun aplikasi Android untuk layanan pemesanan barang?
2. Bagaimana membangun aplikasi layanan pemesanan barang berdasarkan kategori barang?
3. Bagaimana sistem menampilkan status pemesanan secara *real time*?

1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan, yaitu sebagai berikut :

1. Aplikasi yang dibuat adalah aplikasi berbasis perangkat bergerak untuk telepon pintar Android dengan versi minimal 4.1 Jelly Bean.
2. Aplikasi diujikan dengan data barang yang ada di 4 gudang di kota Surabaya.
3. Pengguna aplikasi adalah petugas yang telah terdaftar dalam sistem pergudangan.

4. Tidak adanya menu pendaftaran pengguna dalam aplikasi.

1.4 Tujuan

Tujuan tugas akhir ini bertujuan untuk :

1. Membangun aplikasi Android untuk layanan pemesanan barang.
2. Membangun aplikasi layanan pemesanan barang berdasarkan kategori barang.
3. Menampilkan status pemesanan secara *real time*.

1.5 Manfaat

Dengan dibuatnya tugas akhir ini diharapkan bisa memberikan manfaat yang besar pada pengguna khususnya dinas kota Surabaya dalam melakukan pemesanan barang dan dapat dilihat status pemesanannya.

1.6 Metodologi

Tahapan-tahapan yang dilakukan dalam pengerjaan tugas akhir ini adalah sebagai berikut:

1. Studi literatur

Pada tahap ini dilakukan pengumpulan dan penggalian informasi dan literatur yang diperlukan dalam perancangan dan implementasi aplikasi yang akan dibangun. Literatur yang digunakan adalah sebagai berikut :

- a. Alur proses pemesanan barang di dinas kebersihan dan pertamanan kota Surabaya.
- b. Pengembangan aplikasi pada perangkat bergerak berbasis Android menggunakan kakas bantu Android Studio dan Android SDK (*Software Development Kit*).
- c. Java
- d. MySQL
- e. *Web Service*

- f. JSON
- g. Volley Library
- h. Json Web Token

2. Analisis dan Perancangan sistem

Pada tahap ini dilakukan analisa awal dan mendefinisikan kebutuhan sistem untuk mengetahui masalah yang sedang dihadapi. Dari proses tersebut selanjutnya dirumuskan rancangan sistem yang dapat memberi pemecahan masalah tersebut. Langkah-langkah pada tahap ini adalah sebagai berikut:

- a. Analisis aktor yang terlibat didalam sistem.
- b. Perancangan diagram *use case*, yang merupakan analisis kebutuhan pada aplikasi yang akan dibangun.
- c. Analisis kebutuhan non-fungsional.
- d. Perancangan sistem komunikasi antara server dengan perangkat Android.
- e. Perancangan antarmuka pada aplikasi Android dan Web.

3. Implementasi

Pada tahap ini dilakukan pembuatan perangkat lunak yang merupakan implementasi dari rancangan yang telah dibuat sebelumnya. Sistem yang dibuat berpedoman pada rancangan yang telah dibuat pada proses perancangan dan analisis sistem.

Perincian tahap ini adalah sebagai berikut:

- a. Implementasi antarmuka aplikasi Android dan Web.
- b. Implementasi pemesanan barang.
- c. Implementasi komunikasi antara *server* dengan aplikasi.

4. Pengujian dan Evaluasi

Pada tahap ini akan dilakukan pengujian terhadap perangkat lunak menggunakan skenario yang telah dipersiapkan

sebelumnya. Pengujian dan evaluasi perangkat dilakukan untuk mencari masalah yang mungkin timbul, mengevaluasi jalannya program, dan mengadakan perbaikan jika ada kekurangan. Tahapan-tahapan dari pengujian yang akan dilakukan adalah sebagai berikut :

- a. Uji coba pemesanan barang.
- b. Uji coba implementasi komunikasi server dengan aplikasi, baik aplikasi Android maupun Web.
- c. Uji coba performa.
- d. Uji coba kepada pengguna.

5. Penyusunan Buku Tugas Akhir

Pada tahap ini melakukan pendokumentasian dan laporan dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan, dan hasil-hasil yang telah didapatkan selama pengerjaan tugas akhir. Buku tugas akhir ini bertujuan untuk memberikan gambaran dari pengerjaan tugas akhir ini dan diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut.

1.7 Sistematika Penulisan Laporan Tugas Akhir

Pendokumentasian dan laporan dari seluruh konsep, dasar teori, implementasi, proses yang telah dilakukan, dan hasil-hasil yang telah didapatkan selama pengerjaan tugas akhir. Buku tugas akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan tugas akhir ini dan diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan pengembangan lebih lanjut.

Secara garis besar, buku tugas akhir terdiri atas beberapa bagian seperti berikut ini :

Bab I Pendahuluan

Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan Tugas Akhir. Selain itu perumusan masalah, batasan masalah, metodologi yang

digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

Bab II Tinjauan Pustaka

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan Tugas Akhir ini. Dasar Teori yang digunakan adalah pembangunan aplikasi secara umum.

Bab III Analisis dan Perancangan

Bab ini berisi tentang rancangan sistem yang akan dibangun dan disajikan dalam bentuk diagram aktivitas. Fungsi utama yang akan dibuat pada tugas akhir ini meliputi pemesanan barang, status pemesanan, histori pemesanan dan memperbarui data diri.

Bab IV Implementasi

Bab ini membahas mengenai implementasi dari rancangan yang telah dibuat pada bab sebelumnya. Penjelasan berupa bentuk sumber kode beserta penjelasannya.

Bab V Pengujian dan Evaluasi

Bab ini menjelaskan mengenai kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari perangkat lunak yang telah dibuat sesuai dengan data yang diujikan. Pengujian akan dilakukan dengan menggunakan metode *black box*.

Bab VI Kesimpulan dan Saran

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang telah dilakukan dan

saran untuk pengembangan perangkat lunak kedepannya.

Daftar Pustaka

Merupakan daftar referensi yang digunakan untuk mengembangkan tugas akhir.

Lampiran

Merupakan bab tambahan yang berisi kode-kode sumber yang penting pada aplikasi ini.

[Halaman ini sengaja dikosongkan]

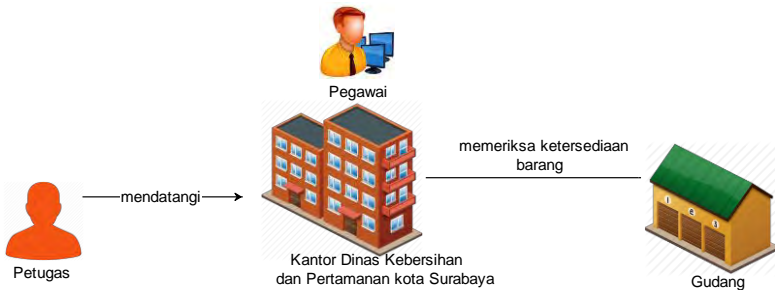
BAB II

TINJAUAN PUSTAKA

Bab ini berisi penjelasan teori-teori yang berkaitan dengan masalah yang diajukan pada pengimplementasian program. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap program yang dibuat dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

2.1 Penelitian Terkait

Pada subbab ini akan dijelaskan tentang penelitian terkait domain permasalahan yang diangkat dalam tugas akhir ini. Permasalahan yang ada adalah proses pemesanan barang di dinas kebersihan dan pertamanan kota Surabaya yang dinilai kurang efisien. Pemesanan barang di dinas kebersihan dan pertamanan saat ini masih dilakukan secara manual. Alur pemesanan barang saat ini dapat dilihat pada Gambar 2.1.



Gambar 2.1 Alur Pemesanan Barang

Petugas harus mendatangi kantor dinas kebersihan dan pertamanan untuk melakukan permohonan pemesanan barang. Pegawai gudang akan mencatat data pemesan dan barang yang ingin dipesan. Pegawai akan memeriksa terlebih dahulu apakah barang yang diminta berada di gudang atau tidak. Selain itu

pegawai memeriksa barang yang tersedia apakah mencukupi dengan jumlah yang diminta petugas. Jika barang yang diminta tersedia, pemesan akan mendapatkan surat bukti pemesanan barang. Untuk mendapatkan barang yang diminta, petugas harus menunggu setidaknya satu minggu setelah menerima surat bukti pemesanan barang[1]. Selain itu petugas juga harus terus berkoordinasi dengan pegawai untuk menanyakan status pemesanannya.

A. Aplikasi Sejenis

Sebelum merancang dan mengimplementasikan sistem, meninjau aplikasi serupa perlu dilakukan guna menentukan standarisasi dalam aplikasi pemesanan barang. Aplikasi yang akan dibuat adalah aplikasi yang digunakan untuk sebuah perusahaan atau instansi sehingga aplikasi yang ditinjau juga yang digunakan dalam skala kecil bukan untuk umum. Aplikasi order barang yang ditinjau adalah aplikasi yang digunakan untuk sebuah instansi dalam melakukan penjualan barang [2]. Nama aplikasi ini adalah “Zahir POS”, aplikasi ini dapat menampilkan barang pada menu produk dan dapat melakukan penjualan barang. Aplikasi ini dapat dilihat pada Gambar 2.2. Dan untuk menu penjualan dapat dilihat pada Gambar 2.3 terdapat informasi gambar, stok, dan harga dari barang. Menu pembayaran dapat dilihat pada Gambar 2.4, terdapat informasi keranjang belanja dan total pembayaran dari pemesanan barang yang telah dilakukan.

Tabel 2.1 Perbandingan Aplikasi Sejenis

No	Fitur	Aplikasi Zahir POS	Aplikasi yang akan dibuat
1	<i>Login</i>	v	v

2	Pemesanan Barang	v	v
3	Kategori Barang		v
4	Stok Barang	v	v
5	Harga Barang	v	
6	Status Pemesanan		v
7	Pembayaran	v	
8	Riwayat Pemesanan	v	v
9	<i>Update</i> Profil		v
10	Kontak	v	v



Gambar 2.2 Antarmuka Aplikasi Zahir POS



Gambar 2.3 Antarmuka Penjualan



Gambar 2.4 Antarmuka Pembayaran

2.2 Android

Android adalah salah satu sistem operasi *open source* berbasis Linux yang ada pada perangkat bergerak yang dioperasikan pada jenis *smartphone*. Awalnya sistem operasi ini dikembangkan oleh Android, Inc dan kemudian pada tahun 2005 dibeli oleh perusahaan raksasa Google, Inc. Sistem operasi ini dirilis tahun 2007 dan saat ini merajai pasar penjualan perangkat bergerak, sehingga menggeser penjualan perangkat bergerak dengan sistem operasi yang lebih lama seperti Java, Symbian, dan Windows Mobile. Pengembangan aplikasi pada perangkat bergerak berbasis Android menggunakan bahasa pemrograman Java. Hingga saat ini, jenis sistem operasi yang dikembangkan pada Android ada beberapa macam, namun yang digunakan untuk tugas akhir ini minimum versi Jellybean (versi 4.1-4.3).

Sistem operasi Android sampai saat ini terus dikembangkan, bahkan sistem operasi ini tidak hanya digunakan pada perangkat bergerak saja, namun perangkat lain yang pada awal pengembangannya tidak ada kaitan langsung dengan perangkat bergerak seperti produk televisi, yaitu Smart TV dan kamera digital, yaitu Galaxy Camera yang dirilis oleh perusahaan raksasa asal Korea, Samsung. Tidak hanya televisi dan kamera digital, bahkan baru-baru ini Google mengembangkan sistem operasi Android yang ditanamkan pada produk mereka, yaitu kacamata pintar Google Glass. Selain itu dikembangkan juga jam tangan pintar yang dapat mengendalikan perangkat bergerak seperti yang dikembangkan oleh Samsung yaitu Galaxy Gear dan Sony dengan produknya Smart Watch Sony [3].

2.3 Java

Java adalah bahasa pemrograman yang dapat dijalankan di berbagai komputer termasuk telepon genggam. Bahasa ini awalnya dibuat oleh James Gosling saat masih bergabung di Sun Microsystems saat ini merupakan bagian dari Oracle dan dirilis

tahun 1995. Bahasa ini banyak mengadopsi sintaksis yang terdapat pada C dan C++ namun dengan sintaksis model objek yang lebih sederhana serta dukungan rutin-rutin atas bawah yang minimal. Aplikasi-aplikasi berbasis java umumnya dikompilasi ke dalam *p-code (bytecode)* dan dapat dijalankan pada berbagai Mesin Virtual Java (JVM). Java merupakan bahasa pemrograman yang bersifat umum/non-spesifik (*general purpose*), dan secara khusus didisain untuk memanfaatkan dependensi implementasi seminimal mungkin. Karena fungsionalitasnya yang memungkinkan aplikasi java mampu berjalan di beberapa platform sistem operasi yang berbeda, java dikenal pula dengan slogannya, "Tulis sekali, jalankan di mana pun". Saat ini java merupakan bahasa pemrograman yang paling populer digunakan, dan secara luas dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi berbasis web.

Bahasa pemrograman Java memiliki kelebihan sebagai berikut:

1. *Multiplatform*. Kelebihan utama dari Java ialah dapat dijalankan di beberapa platform / sistem operasi komputer, sesuai dengan prinsip tulis sekali, jalankan di mana saja. Dengan kelebihan ini pemrogram cukup menulis sebuah program Java dan dikompilasi (diubah, dari bahasa yang dimengerti manusia menjadi bahasa mesin / *bytecode*) sekali lalu hasilnya dapat dijalankan di atas beberapa platform tanpa perubahan. Kelebihan ini memungkinkan sebuah program berbasis java dikerjakan diatas operating system Linux tetapi dijalankan dengan baik di atas Microsoft Windows. Platform yang didukung sampai saat ini adalah Microsoft Windows, Linux, Mac OS dan Sun Solaris. Penyebabnya adalah setiap sistem operasi menggunakan programnya sendiri-sendiri (yang dapat diunduh dari situs Java) untuk meninterpretasikan *bytecode* tersebut.

2. OOP (*Object Oriented Programming* - Pemrogram Berorientasi Objek).
3. *Class Library* Yang Lengkap, Java terkenal dengan kelengkapan *library*/perpustakaan (kumpulan program program yang disertakan dalam pemrograman java) yang sangat memudahkan dalam penggunaan oleh para pemrogram untuk membangun aplikasinya. Kelengkapan perpustakaan ini ditambah dengan keberadaan komunitas Java yang besar yang terus menerus membuat perpustakaan-perpustakaan baru untuk melingkupi seluruh kebutuhan pembangunan aplikasi.
4. Bergaya C++, memiliki sintaks seperti bahasa pemrograman C++ sehingga menarik banyak pemrogram C++ untuk pindah ke Java. Saat ini pengguna Java sangat banyak, sebagian besar adalah pemrogram C++ yang pindah ke Java. Universitas-universitas di Amerika Serikat juga mulai berpindah dengan mengajarkan Java kepada murid-murid yang baru karena lebih mudah dipahami oleh murid dan dapat berguna juga bagi mereka yang bukan mengambil jurusan komputer.
5. Pengumpulan sampah otomatis, yaitu memiliki fasilitas pengaturan penggunaan memori sehingga para pemrogram tidak perlu melakukan pengaturan memori secara langsung (seperti halnya dalam bahasa C++ yang dipakai secara luas).

Selain kelebihan diatas, bahasa pemrograman Java juga memiliki beberapa kekurangan:

1. Masih ada beberapa hal yang tidak kompatibel antara platform satu dengan platform lain. Untuk J2SE, misalnya SWT-AWT bridge yang sampai sekarang tidak berfungsi pada Mac OS X.
2. Mudah didekompilasi. Dekompilasi adalah proses membalikkan dari kode jadi menjadi kode sumber. Ini dimungkinkan karena kode jadi Java merupakan *bytecode*

yang menyimpan banyak atribut bahasa tingkat tinggi, seperti nama-nama kelas, metode, dan tipe data. Hal yang sama juga terjadi pada Platform Microsoft .NET. Dengan demikian, algoritma yang digunakan program akan lebih sulit disembunyikan dan mudah dibajak/*direverse-engineer*.

Penggunaan memori yang banyak. Penggunaan memori untuk program berbasis Java jauh lebih besar daripada bahasa tingkat tinggi generasi sebelumnya seperti C/C++ dan Pascal (lebih spesifik lagi, Delphi dan Object Pascal) [3]. Biasanya ini bukan merupakan masalah bagi pihak yang menggunakan teknologi terbaru (karena tren memori terpasang makin murah), tetapi menjadi masalah bagi mereka yang masih harus berkuat dengan mesin komputer berumur lebih dari 4 tahun.

2.4 Android SDK

Android SDK (*Software Development Kit*) merupakan kakas bagi para pengembang yang ingin mengembangkan aplikasi berbasis google android. Android SDK mencakup seperangkat alat pengembangan yang komprehensif. Android SDK terdiri dari *debugger*, *libraries*, *handset emulator*, dokumentasi, contoh kode, dan tutorial. Saat ini Android sudah mendukung arsitektur x86 pada Linux (distribusi Linux apapun untuk desktop modern), Mac OS X 10.4.8 atau lebih, Windows XP atau di atasnya. Persyaratannya mencakup JDK, Apache Ant dan Python 2.2 atau yang lebih baru. IDE yang didukung secara resmi adalah Eclipse 3.2 atau lebih dengan menggunakan plugin Android Development Tools (ADT), dengan ini pengembang dapat menggunakan teks editor untuk mengedit file Java, yang digunakan untuk melakukan eksekusi baris kode dan XML yang digunakan untuk melakukan desain antarmuka pengguna serta menggunakan peralatan *command line* untuk menciptakan, membangun, melakukan *debug* aplikasi Android dan

pengendalian perangkat Android (misalnya, reboot, menginstal paket perangkat lunak dengan jarak jauh).

Android SDK telah dirilis pada tanggal 12 November 2007. Dan pada tanggal 15 Juli 2008 tim Android Developer Challenge sengaja mengirimkan email ke semua pendatang di Android Developer Challenge untuk mengumumkan bahwa rilis SDK terbaru telah tersedia pada halaman download pribadi. Email tersebut juga ditujukan kepada pemenang Android Developer Challenge putaran pertama. Sebuah pernyataan bahwa Google telah menyediakan rilis SDK terbaru untuk beberapa pengembang dan bukan untuk orang lain [4].

Pada tanggal 18 Agustus 2008, Android SDK 0.9 beta dirilis. Rilis ini menyediakan API yang diperbarui dan diperluas, perbaikan pada alat-alat pengembangan dan desain terbaru untuk layar awal. Petunjuk untuk meng-upgrade SDK sudah tersedia pada rilis sebelumnya. Pada tanggal 23 September 2008, Android 1.0 SDK telah dirilis. Pada tanggal 9 Maret 2009, Google merilis versi 1.1 untuk telepon seluler Android. Rilis terbaru tersebut termasuk dukungan untuk pencarian dengan suara, harga aplikasi, perbaikan jam alarm, perbaikan pengiriman gmail, perbaikan surat pemberitahuan dan peta.

Pada pertengahan Mei 2009, Google merilis versi 1.5 (Cupcake) pada sistem operasi Android dan SDK. Pembaruan ini termasuk banyak fitur baru seperti perekaman video, dukungan untuk bluetooth, sistem keyboard pada layar dan pengenalan suara. Rilis ini juga membuka AppWidget *framework* kepada para pengembang yang memungkinkan orang untuk membuat widget sendiri pada halaman home. Pada September 2009 versi 1.6 (Donut) dirilis yang menampilkan hasil pencarian yang lebih baik dan penggunaan indikator baterai. Hingga saat ini, Android SDK terus dikembangkan hingga pengembang dapat melakukan pengembangan perangkat lunak dengan Android versi terbaru (KitKat Android versi 4.0).

2.5 Kakas Kerja Android Studio

Android Studio adalah sebuah IDE (*Integrated Development Environment*) untuk perangkat lunak berbasis Android. Pada tugas akhir ini kakas kerja Android Studio yang digunakan adalah 1.5.1.

2.6 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL atau DBMS yang *multithread*, *multi-user*, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, yang memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah: David Axmark, Allan Larsson, dan Michael "Monty" Widenius.

MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (*General Public License*). Setiap pengguna dapat secara bebas dan gratis dalam menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya; SQL (*Structured Query Language*). SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis [5].

Kehandalan suatu sistem basis data (DBMS) dapat diketahui dari cara kerja optimasinya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai penyedia basis data, MySQL mendukung operasi basis data transaksional maupun operasi basis data non-transaksional. Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak penyedia basis data kompetitor lainnya. Namun demikian pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi *blogging* berbasis web (Wordpress atau Blogspot), CMS (*Content Management System*), dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basisdata transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional.

MySQL memiliki beberapa kelebihan, antara lain:

1. Portabilitas. MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.
2. Perangkat lunak sumber terbuka. MySQL didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi GPL sehingga dapat digunakan secara gratis.
3. *Multi-user*. MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. '*Performance tuning*', MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. Ragam tipe data. MySQL memiliki ragam tipe data yang sangat kaya, seperti *signed / unsigned* integer, float, double, char, text, date, timestamp, dan lain-lain.
6. Perintah dan Fungsi. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (*query*).
7. Keamanan. MySQL memiliki beberapa lapisan keamanan seperti level *subnetmask*, nama *host*, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.
8. Skalabilitas dan Pembatasan. MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (*records*) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.
9. Konektivitas. MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).
10. Lokalisasi. MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa.
11. Antar Muka. MySQL memiliki antar muka (interface) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (*Application Programming Interface*).
12. Klien dan kakas. MySQL dilengkapi dengan berbagai kakas (*tools*) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk *online*.

Struktur tabel. MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya semacam PostgreSQL ataupun Oracle.

2.7 Volley Library

Volley adalah sebuah library yang digunakan untuk membantu pertukaran data dari server dengan client. Volley membuat pertukaran data menjadi lebih mudah dan lebih cepat. Secara default semua jaringan volley menghasilkan *asynchronous* sehingga tidak perlu khawatir menggunakan AsyncTask [6]. Volley dilengkapi dengan banyak fitur, diantaranya adalah :

1. Antrian permintaan dan prioritas
2. Efektif *cache* dan manajemen memori
3. Perluasan dan kustomisasi *library* untuk kebutuhan pengguna
4. Membatalkan permintaan.

2.8 JSON

JSON adalah singkatan dari *Java Script Object Notation*, yaitu sebuah format untuk pertukaran data. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman Java Script, Standar ECMA-262 Edisi ke-3 Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data. JSON terbuat dari dua struktur:

- Kumpulan pasangan nama/nilai. Pada beberapa bahasa, hal ini dinyatakan sebagai objek (*object*), rekaman (*record*), struktur (*struct*), kamus (*dictionary*), tabel hash (*hash table*), daftar berkunci (*keyed list*), atau *associative array*.
- Daftar nilai terurutkan (*an ordered list of values*). Pada kebanyakan bahasa, hal ini dinyatakan sebagai larik (*array*), vektor (*vector*), daftar (*list*), atau urutan (*sequence*).

Struktur-struktur data ini disebut sebagai struktur data universal. Pada dasarnya, semua bahasa pemrograman modern mendukung struktur data ini dalam bentuk yang sama maupun berlainan. Hal ini pantas disebut demikian karena format data mudah dipertukarkan dengan bahasa-bahasa pemrograman yang juga berdasarkan pada struktur data ini [7]. Contoh format JSON yang digunakan adalah sebagai berikut.

```
{
  "content": [
    {
      "id_req": "1",
      "nama_barang": "Lampu"
    },
    {
      "id_req": "2",
      "nama_barang": "Lampu"
    }
  ]
}
```

Kode Sumber 2.1 Format JSON

2.9 JSON Web Token

JSON web token adalah sebuah URL yang dikemas dengan aman untuk mengklaim transaksi anatar 2 buah party. Klaimnya sendiri di encode menjadi JSON Object sebagai tanda tangan digital menggunakan JWS (*JSON Web Signature*). Struktur JWT dibagi menjadi 3 bagian yaitu :

1. Header

```
<base64-encoded header>
```

Header memiliki 2 atribut, typ dan alg dengan masing-masing tipe data String. Typ adalah tipe token dengan default “JWT” dan alg adalah algoritma yang digunakan oleh *signature*.

Contoh dari header adalah sebagai berikut :

```
{
  "typ" : "JWT",
  "alg" : "HS256"
}
```

Kode Sumber 2.2 Header JSON Web Token

2. Claim

```
<base64-encoded claims>
```

Claim adalah sebuah pesan yang digunakan sebagai keamanan. Claim memiliki 6 atribut yaitu iss, iat, exp, qsh, sub, dan aud. Masing-masing atribut memiliki tipe data. Iss digunakan untuk mengidentifikasi klien. Iat digunakan untuk membuat tanggal token dibuat. Exp adalah masa berlaku token. Qsh adalah *query Hash* dengan tipe data String. Sub adalah subjek token, dan aud audien token yang berupa informasi publik. Contoh dari claim adalah sebagai berikut :

```
{
  "iss": "jira:1314039",
  "iat": 1300819370,
  "exp": 1300819380,
  "qsh":
    "8063ff4ca1e41df7bc90c8ab6d0f6207d491cf6d
    ad7c66ea797b4614b71922e9"
}
```

Kode Sumber 2.3 Claim JSON Web Token

3. Signature

```
<base64-encoded signature>
```

Pada bagian ini data diberi sebuah tanda tangan digital guna memastikan hak akses yang dimiliki. JSON Object yang telah di bentuk pada saat claim akan diberi tanda tangan digital menggunakan JWS(*JSON Web Signature*).

BAB III

ANALISIS DAN PERANCANGAN

Pada bab ini akan dijelaskan analisis dan perancangan perangkat lunak yang dibuat. Tahapan mulai dari analisis sistem, perancangan sistem, perancangan perangkat lunak, dan implementasi perangkat lunak. Pada bab ini juga akan dijelaskan mengenai gambaran umum setiap proses utama program dalam diagram alir beserta penjelasannya. Pendekatan yang dibuat dalam perancangan ini adalah pendekatan berorientasi objek. Perancangan direpresentasikan dengan diagram UML (*Unified Modelling Language*).

3.1 Analisis

Tahap analisis ini terbagi menjadi beberapa bagian antara lain, ranah permasalahan dan deskripsi umum perangkat lunak, arsitektur perangkat lunak, dan spesifikasi kebutuhan perangkat lunak. Berikut penjabaran bagian-bagian tahap analisis.

3.1.1 Domain Permasalahan

Kebersihan dan keindahan suatu kota adalah suatu hal yang harus diperhatikan. Jika di suatu kota terdapat kerusakan dan kesalahan infrastruktur, hendaknya pemerintah kota tersebut bergerak cepat untuk menanganinya. Tentunya dengan melewati prosedur-prosedur yang sudah ada. Seperti yang terjadi di kota Surabaya, khususnya dinas kebersihan dan pertamanan. Dinas ini secara garis besar masih mengandalkan peralatan manual untuk menanggapi kerusakan dan kesalahan infrastruktur kota. Jenis infrastruktur yang ditangani oleh dinas kebersihan dan pertamanan dibagi menjadi 4 kategori yaitu, penerangan jalan umum, alat tulis kantor, tanaman, dan peralatan kebersihan. Kerusakan infrastuktur yang biasanya terjadi adalah kurangnya tempat sampah di sekitar taman kota, selain itu gerobak sampah

di berbagai wilayah Surabaya yang kondisinya sudah tidak layak pakai. Tempat sampah dan gerobak sampah adalah salah satu infrastuktur dalam kategori peralatan kebersihan. Jika terdapat kerusakan suatu infrastuktur petugas lapangan dari dinas kebersihan dan pertamanan harus melaporkan dan memperbaiki atau mengganti infrastuktur tersebut. Alur untuk mengganti infrastuktur yang rusak saat ini masih menggunakan cara manual, yaitu petugas lapangan harus ke kantor terlebih dahulu untuk menginformasikan kerusakan infrastuktur. Kemudian, petugas diharuskan mengisi surat permintaan barang untuk perbaikan. Proses ini sudah sangat tidak efektif terlebih pegawai dinas harus melihat ketersediaan barang yang diminta di gudang. Jika barang yang diminta ada, petugas akan mencatat pemesanan barang yang diminta. Petugas lapangan menunggu barang yang dipesan apakah sudah dikirim atau belum, dengan datang ke tempat yang membutuhkan infrastuktur tersebut. Koordinasi pengiriman barang antara petugas lapangan dan pegawai dinas masih menggunakan telepon maupun sms sehingga rawan akan *miss-communication*. Jika proses ini tetap dijalankan akan berdampak pada kesejahteraan masyarakat dimana pelayanan pemerintah yang lamban terhadap masalah kebersihan dan pertamanan kota.

3.1.2 Deskripsi Umum

Berdasarkan permasalahan yang ada pada pembahasan domain permasalahan, solusi aplikasi yang ditawarkan adalah pembuatan aplikasi yang dipasang di *smartphone*. Aplikasi ini dapat melakukan pemesanan barang untuk penanganan masalah infrastuktur yang berkaitan dengan kebersihan dan pertamanan kota Surabaya. Dengan menampilkan jumlah barang yang tersedia di gudang, pemesanan barang akan semakin cepat. Setelah memesan barang aplikasi juga akan menampilkan status pemesanan barang sampai status pemesanan selesai dikirim. Aplikasi yang akan dibuat juga dapat digunakan untuk melihat riwayat pemesanan barang. Untuk menangani pemesanan aplikasi

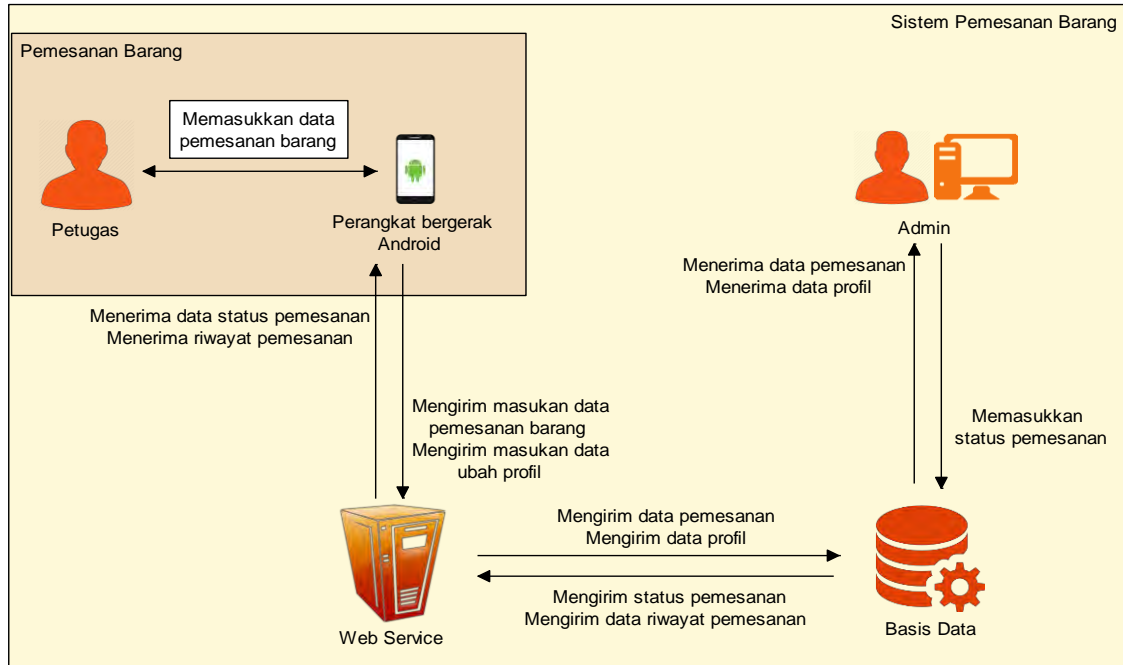
pegawai dinas juga dapat manajemen pemesanan serta pengguna aplikasi.

3.1.3 Aktor

Pada sistem yang akan dibuat di tugas akhir ini, aktor yang menjadi pengguna sistem adalah pegawai gudang di divisi administrasi gudang dan petugas dinas kebersihan dan pertamanan di divisi turun jalan atau petugas lapangan. Petugas dinas kebersihan dan pertamanan menggunakan sistem yang dibangun pada perangkat bergerak berbasis Android yang digunakan untuk memesan barang dan pegawai gudang adalah administrator web yang menggunakan sistem aplikasi web untuk manajemen pemesanan dan manajemen pengguna aplikasi.

3.2 Arsitektur Sistem

Dalam aplikasi yang dibuat pada tugas akhir ini, sistem terdiri dari dua aplikasi, yaitu aplikasi berbasis android pada perangkat bergerak dan aplikasi web. Pada aplikasi android, digunakan petugas untuk melakukan pemesanan barang. Petugas juga dapat melihat status pemesanan dan melihat riwayat pemesanan yang telah dilakukan oleh petugas. Sedangkan aplikasi web digunakan pegawai untuk manajemen data pemesanan, data pengguna, dan Melihat semua riwayat pemesanan barang. Agar pertukaran data antara aplikasi perangkat bergerak dan aplikasi web dapat saling terhubung, maka diperlukan sebuah web service dan database untuk menyimpan segala data yang dibutuhkan untuk melakukan pemesanan barang. Arsitektur sistem dapat dilihat pada Gambar 3.1.



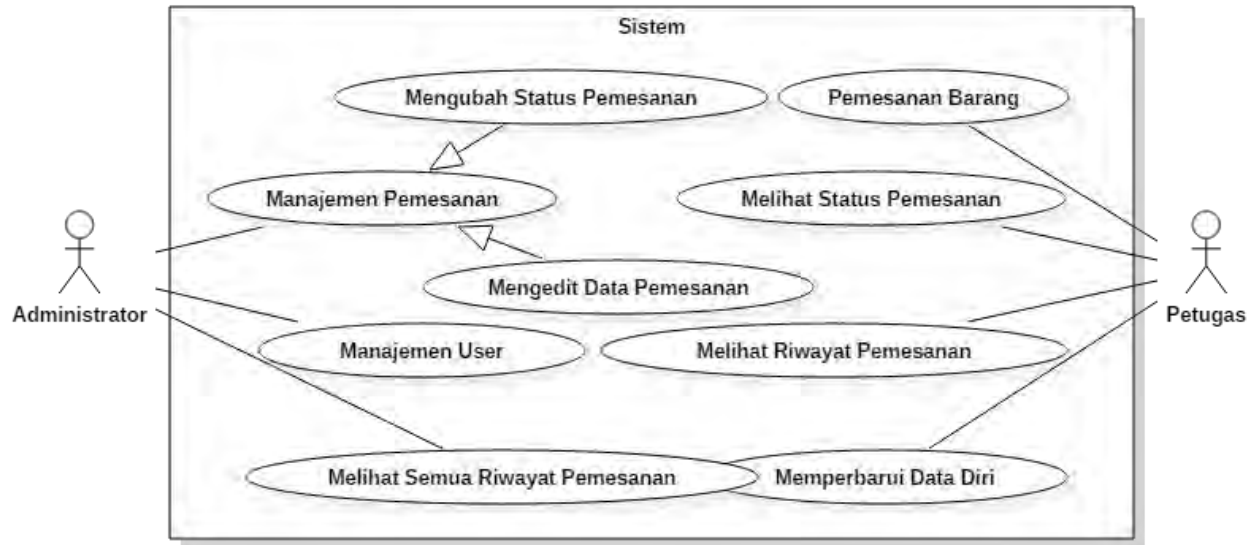
Gambar 3.1 Arsitektur Sistem

3.3 Kasus Penggunaan

Pada subbab ini akan dijelaskan kasus penggunaan yang dibutuhkan pada sistem sesuai dengan analisa yang telah dilakukan. Diagram kasus penggunaan dapat dilihat pada Gambar 3.2 dan kode kasus penggunaan ada pada Tabel 3.1.

Terdapat 7 kasus penggunaan yang dibutuhkan pada sistem. Kasus penggunaan pemesanan barang adalah kasus penggunaan yang utama dimana aplikasi dapat melakukan pemesanan barang. Kasus penggunaan melihat status pemesanan dibutuhkan sistem pada saat pemesanan telah dilakukan. Pemesanan akan memiliki status pemesanan disetiap transaksi pemesanan. Status pemesanan dibagi menjadi 4 status, yaitu pesan, ok, dikirim, dan selesai. Setiap pemesanan yang telah dilakukan status pemesanan akan otomatis masuk pada status pemesanan 1 yaitu pesan. Kasus penggunaan melihat riwayat pemesanan dibutuhkan sistem untuk melihat riwayat pemesanan yang dilakukan oleh pengguna tersebut. Kasus penggunaan memperbarui data diri dibutuhkan sistem untuk memperbarui data diri pengguna aplikasi perangkat bergerak.

Pada aplikasi web terdapat 3 kasus penggunaan, yang pertama kasus penggunaan manajemen pemesanan. Kasus penggunaan ini dibutuhkan sistem untuk mengelola data pemesanan beserta status pemesanannya. Pada kasus penggunaan ini pengguna dapat merubah data pemesanan dan status pemesanannya. Kasus penggunaan Melihat Semua Riwayat Pemesanan dibutuhkan sistem untuk melihat riwayat seluruh pemesanan yang telah dilakukan. Dan yang terakhir adalah kasus penggunaan manajemen *user*, dibutuhkan sistem untuk mengelola pengguna aplikasi perangkat bergerak. Kasus penggunaan ini juga dapat menambah pengguna aplikasi, mengedit data pengguna aplikasi, dan menghapus pengguna aplikasi.



Gambar 3.2 Kasus Penggunaan

Tabel 3.1 Kode Kasus Penggunaan

Kode Kasus Penggunaan	Kasus Penggunaan
UC-001	Pemesanan Barang
UC-002	Melihat Status Pemesanan
UC-003	Melihat Riwayat Pemesanan
UC-004	Memperbarui Data Diri
UC-005	Manajemen Pemesanan
UC-006	Manajemen <i>User</i>
UC-007	Melihat Semua Riwayat Pemesanan

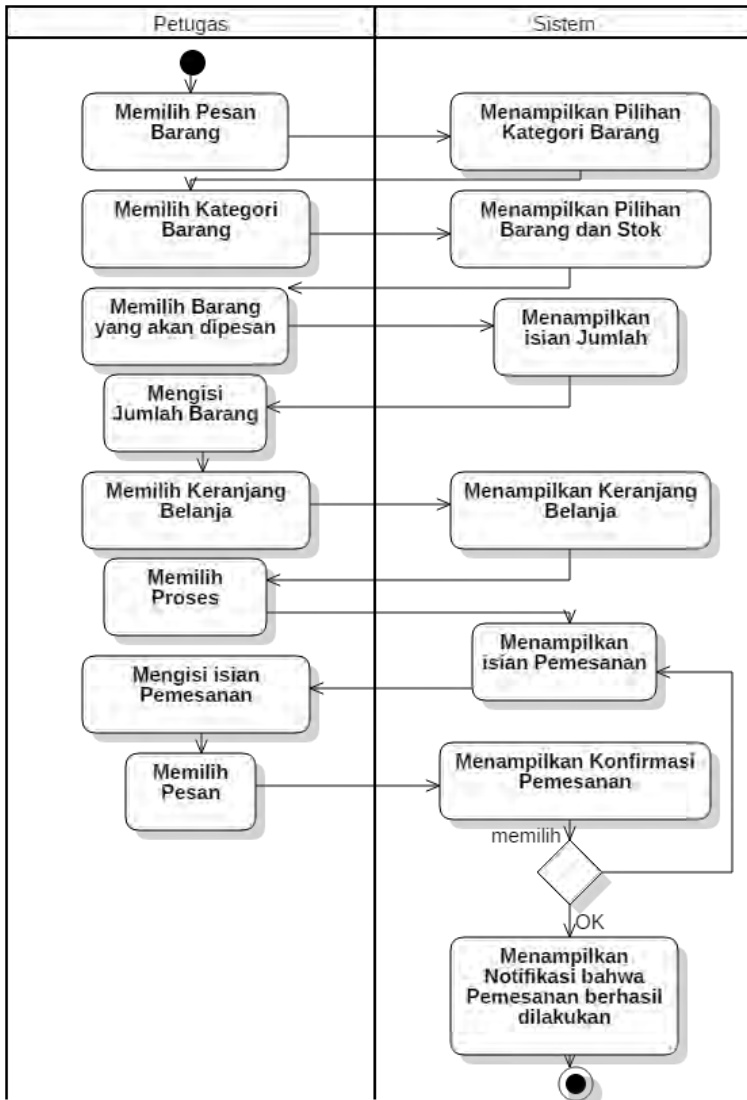
3.3.1 Kasus Penggunaan Pemesanan Barang

Pada kasus penggunaan ini, sistem menampilkan pilihan kategori barang dan kemudian menampilkan pilihan barang-barang beserta stok yang tersedia. Pengguna menambahkan beberapa informasi pemesanan yaitu jumlah barang, informasi alasan pemesanan barang, alamat pengiriman barang, dan tanggal pemesanan. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.2. Diagram aktivitas pemesanan barang dapat dilihat pada Gambar 3.3.

Tabel 3.2 Spesifikasi Pemesanan Barang

Nama	Pemesanan Barang
Kode	UC-001
Deskripsi	Sistem menampilkan pilihan kategori barang, kemudian sistem menampilkan pilihan barang dan stok barang sesuai kategori barang. Kemudian sistem menerima masukan data barang untuk ditampilkan ke langkah selanjutnya yaitu memasukkan jumlah barang. Sistem menampilkan dan menerima kotak masukan yang harus diisi oleh

	pengguna yang nantinya akan disimpan ke basis data.
Tipe	Fungsional
Pemicu	Pengguna menekan “Pesan”
Aktor	Petugas
Kondisi Awal	Pengguna berada pada halaman utama dan memilih pesan barang.
Aliran: - Kejadian Normal	1. Sistem menampilkan pilihan kategori barang 2. Pengguna memilih kategori barang. 3. Sistem menampilkan pilihan barang dan stok yang tersedia berdasarkan kategori barang yang dipilih. 4. Pengguna memilih barang yang ingin dipesan. 5. Sistem menampilkan kotak masukan jumlah barang. 6. Pengguna mengisi jumlah barang. 7. Pengguna memilih masukkan keranjang belanja. 8. Pengguna memilih keranjang belanja 9. Sistem menampilkan daftar keranjang belanja. 10. Pengguna memilih proses. 11. Sistem menampilkan 3 kotak masukan yaitu, alasan pemesanan, alamat pengiriman, dan tanggal pemesanan. 12. Pengguna mengisi kotak isian dengan sesuai. 13. Pengguna memilih Pesan 14. Sistem mengirimkan data pada basis data.
- Kejadian Alternatif	-
Kondisi Akhir	Data masuk pada basis data.



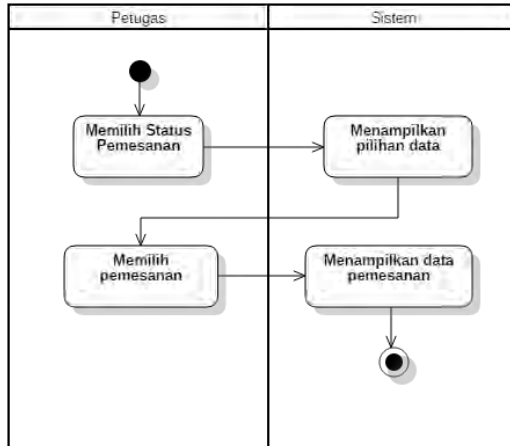
Gambar 3.3 Diagram Aktivitas Pemesanan Barang

3.3.2 Kasus Penggunaan Melihat Status Pemesanan

Pada kasus penggunaan ini, sistem menampilkan status pemesanan pada pemesanan telah dilakukan. Sistem menampilkan pilihan pemesanan, pengguna memilih pemesanan yang ingin dilihat status pemesanannya. Sistem akan menampilkan data pemesanan yang dipilih dan status pemesanannya. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.3. Diagram aktivitas melihat status pemesanan dapat dilihat pada Gambar 3.4.

Tabel 3.3 Spesifikasi Melihat Status Pemesanan

Nama	Melihat Status Pemesanan
Kode	UC-002
Deskripsi	Sistem menampilkan pilihan pemesanan yang telah dilakukan, pengguna dapat melihat status pemesanan dengan menekan salah satu pilihan status pemesanan. Sistem akan menampilkan detail data pemesanan dan status pemesanannya. Status pemesanan ini berupa gambar-gambar.
Tipe	Fungsional
Pemicu	Pengguna menekan “Status Pemesanan”
Aktor	Petugas
Kondisi Awal	Pengguna berada pada halaman lacak barang dan memilih status pemesanan.
Aliran: - Kejadian Normal	1. Sistem menampilkan pilihan pemesanan 2. Pengguna memilih pemesanan 3. Sistem menampilkan data pemesanan beserta status pemesanannya.
- Kejadian Alternatif	-
Kondisi Akhir	Data pemesanan tampil pada sistem.



Gambar 3.4 Diagram Aktivitas Melihat Status pemesanan

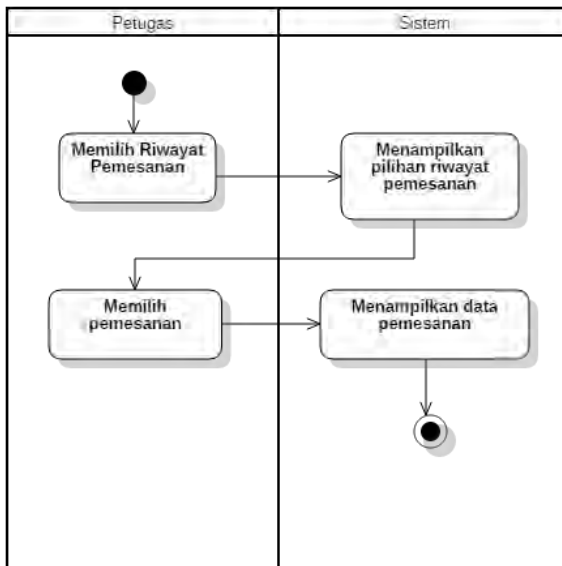
3.3.3 Kasus Penggunaan Melihat Riwayat Pemesanan

Pada kasus penggunaan ini, sistem menampilkan riwayat pemesanan yang telah dilakukan. Sistem menampilkan pilihan pemesanan, pengguna memilih pemesanan yang ingin dilihat. Sistem akan menampilkan data pemesanan yang telah dipilih. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.4. Diagram aktivitas melihat riwayat pemesanan dapat dilihat pada Gambar 3.5.

Tabel 3.4 Spesifikasi Melihat Riwayat Pemesanan

Nama	Melihat Riwayat Pemesanan
Kode	UC-003
Deskripsi	Sistem menampilkan pilihan pemesanan yang telah dilakukan oleh pengguna, pengguna menekan salah satu pilihan riwayat pemesanan. Sistem akan menampilkan detail

	data pemesanan.
Tipe	Fungsional
Pemicu	Pengguna menekan “Riwayat Pemesanan”
Aktor	Petugas
Kondisi Awal	Pengguna berada pada halaman lacak barang dan memilih riwayat pemesanan.
Aliran:	1. Sistem menampilkan pilihan pemesanan
- Kejadian Normal	2. Pengguna memilih pemesanan yang ingin dilihat.
	3. Sistem menampilkan data pemesanan yang dipilih.
- Kejadian Alternatif	-
Kondisi Akhir	Data pemesanan yang dipilih tampil pada sistem.



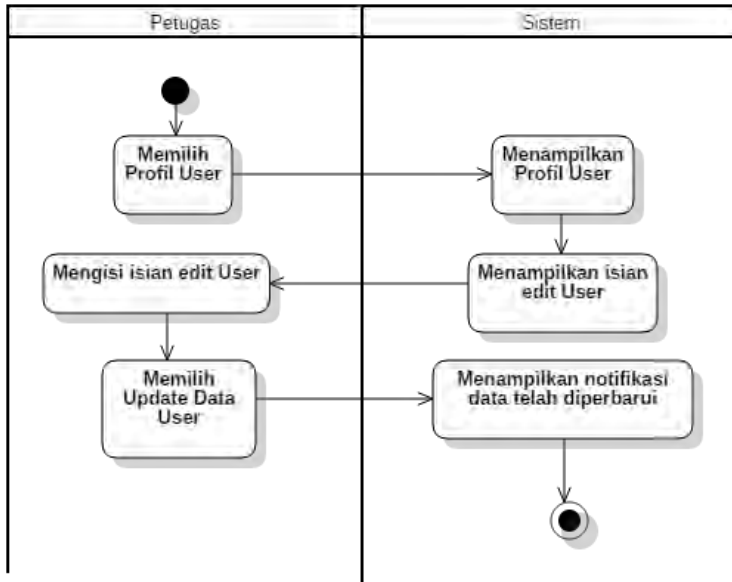
Gambar 3.5 Diagram Aktivitas Melihat Riwayat Pemesanan

3.3.4 Kasus Penggunaan Memperbarui Data Diri

Pada kasus penggunaan ini, pengguna dapat memperbarui data diri. Sistem akan menampilkan data diri yang ada di basis data, jika pengguna ingin memperbarui data diri, pengguna mengisi kotak masukan sesuai dengan data yang ingin diperbarui. Sistem akan mencocokkan *password* pengguna yang baru guna memastikan *password* baru valid. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.5. Diagram aktivitas memperbarui data diri dapat dilihat pada Gambar 3.6.

Tabel 3.5 Spesifikasi Memperbarui Data Diri

Nama	Memperbarui Data Diri
Kode	UC-004
Deskripsi	Sistem menampilkan data pengguna dan pengguna mengisi kotak isian sesuai dengan data yang ingin diubah oleh pengguna. Sistem menerima masukan dari kotak masukkan yang terdapat di halaman ini.
Tipe	Fungsional.
Pemicu	Pengguna menekan “Profil”
Aktor	Petugas
Kondisi Awal	Pengguna berada pada halaman Profil
Aliran: - Kejadian Normal	1. Sistem menampilkan data pengguna 2. Pengguna mengisi kotak masukkan 3. Pengguna menekan Simpan 4. Sistem menerima masukan dari kotak masukkan yang telah diisi.
- Kejadian Alternatif	-
Kondisi Akhir	Data diri disimpan ke dalam basis data.



Gambar 3.6 Diagram Aktivitas Memperbarui Data Diri

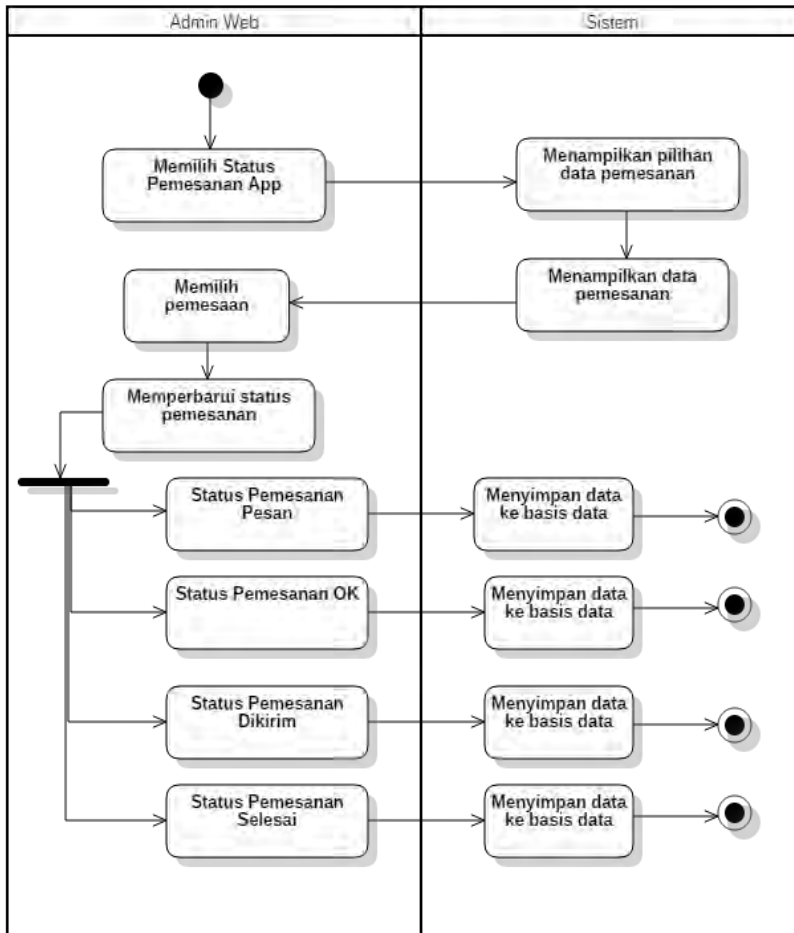
3.3.5 Kasus Penggunaan Manajemen Pemesanan

Setelah kasus penggunaan UC-001 dilakukan, data akan ditampilkan pada kasus penggunaan ini. Sistem akan menampilkan pemesanan yang baru masuk. Pengguna dapat merubah data pemesanan dan status pemesanan. Status pemesanan dibagi menjadi 4 macam, yang pertama adalah pesan dimana pemesanan barang yang masuk akan otomatis berada pada status pertama atau pesan. Status yang kedua adalah ok dimana pemesanan telah disetujui dan dilihat oleh administrator web. Status ketiga adalah dikirim dimana barang yang telah dipesan akan dikirim sesuai alamat pengiriman yang telah diisi pada UC-001. Dan status yang terakhir adalah selesai dimana barang telah dikirimkan dan pemesanan barang telah selesai.

Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.6. Diagram aktivitas manajemen pemesanan dapat dilihat pada Gambar 3.7.

Tabel 3.6 Spesifikasi Manajemen Pemesanan

Nama	Manajemen Pemesanan
Kode	UC-005
Deskripsi	Sistem menampilkan pilihan pemesanan barang yang telah masuk, pengguna dapat mengubah data dan status pemesanannya.
Tipe	Fungsional
Pemicu	Pengguna memilih status pemesanan.
Aktor	Administrator web
Kondisi Awal	Pengguna berada pada halaman Status Pemesanan
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Sistem akan menampilkan pilihan pemesanan 2. Pengguna menekan edit pemesanan 3. Sistem menampilkan detail data pemesanan 4. Pengguna memperbarui data dan status pemesanan. 5. Pengguna memilih Simpan.
- Kejadian Alternatif	<ol style="list-style-type: none"> 2.1 Pengguna menekan detail pemesanan 2.2 Sistem menampilkan detail data pemesanan
Kondisi Akhir	Data pemesanan berhasil disimpan di basis data



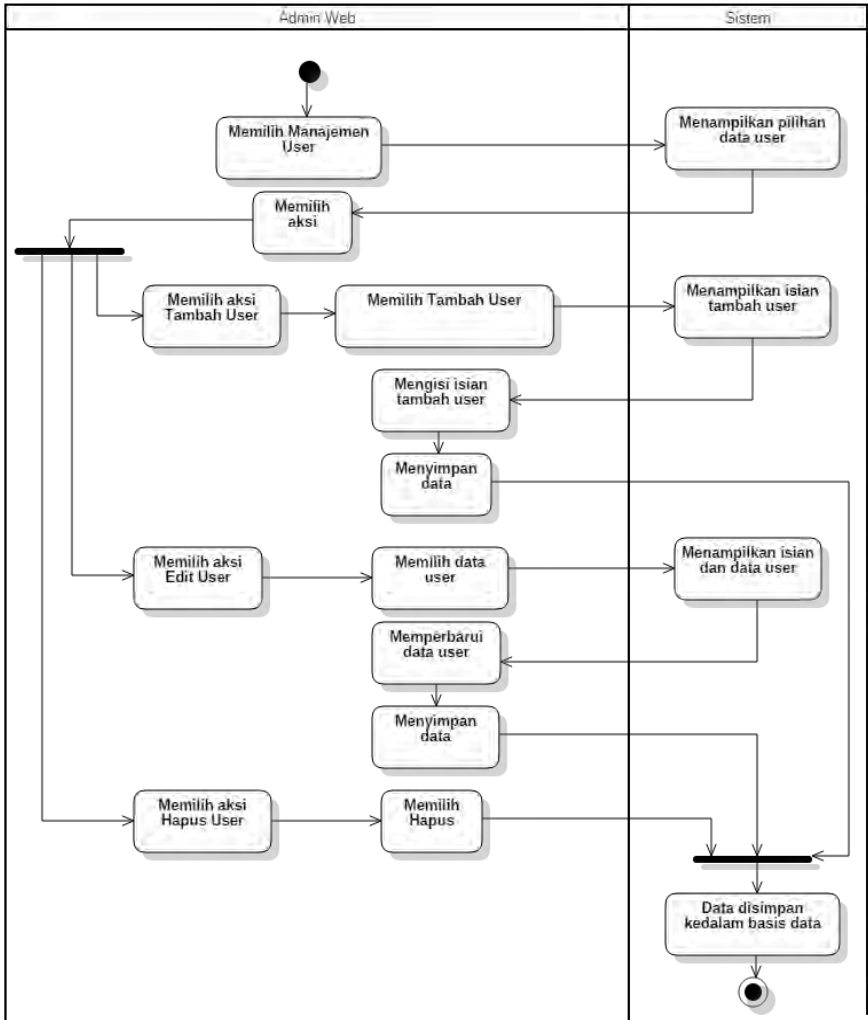
Gambar 3.7 Diagram Aktivitas Manajemen Pemesanan

3.3.6 Kasus Penggunaan Manajemen *User*

Pada kasus penggunaan ini, pengguna dapat manajemen *user* aplikasi perangkat bergerak. Pengguna dapat menambahkan, memperbarui, dan menghapus data pengguna aplikasi. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.7. Diagram aktivitas manajemen *user* dapat dilihat pada Gambar 3.8.

Tabel 3.7 Spesifikasi Manajemen *User*

Nama	Manajemen <i>User</i>
Kode	UC-006
Deskripsi	Sistem dapat menambahkan, memperbarui, dan menghapus data <i>user</i> .
Tipe	Fungsional
Pemicu	Pengguna memilih Manajemen User
Aktor	Administrator web
Kondisi Awal	Pengguna berada pada halaman Manajemen User
Aliran: - Kejadian Normal	<ol style="list-style-type: none"> 1. Sistem menampilkan pilihan <i>user</i> 2. Pengguna memilih <i>user</i> yang akan diubah datanya. 3. Sistem menampilkan data yang telah dipilih. 4. Pengguna menyimpan perubahan yang telah dilakukan 5. Sistem menyimpan data ke dalam basis data.
- Kejadian Alternatif	-
Kondisi Akhir	Data <i>user</i> berhasil disimpan di basis data

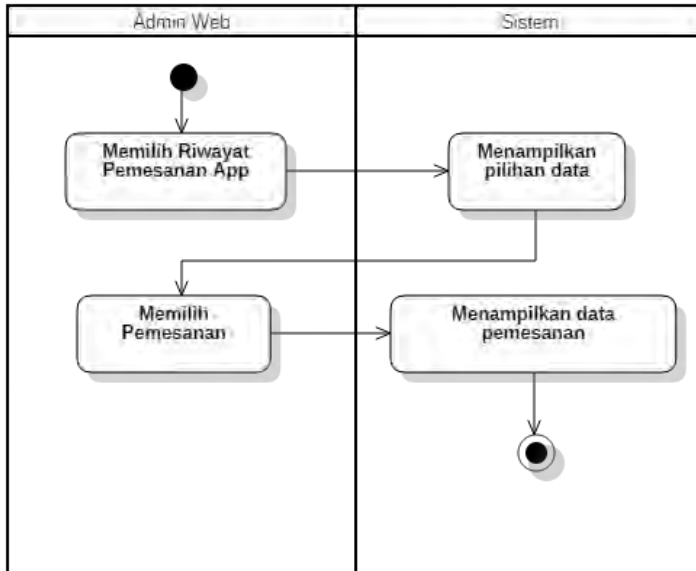
Gambar 3.8 Diagram Aktivitas Manajemen *User*

3.3.7 Kasus Penggunaan Melihat Semua Riwayat Pemesanan

Pada kasus penggunaan ini, sistem menampilkan riwayat pemesanan yang telah dilakukan. Sistem menampilkan pilihan pemesanan, pengguna memilih pemesanan yang ingin dilihat. Sistem akan menampilkan data pemesanan yang telah dipilih. Spesifikasi kasus penggunaan ini dapat dilihat pada Tabel 3.8. Diagram aktivitas melihat semua riwayat pemesanan dapat dilihat pada Gambar 3.9.

Tabel 3.8 Spesifikasi Melihat Semua Riwayat Pemesanan

Nama	Melihat Semua Riwayat Pemesanan
Kode	UC-007
Deskripsi	Sistem menampilkan pilihan pemesanan yang telah dilakukan, pengguna dapat melihat detail pemesanan dengan menekan salah satu pilihan pemesanan. Sistem akan menampilkan data pemesanan.
Tipe	Fungsional
Pemicu	Pengguna memilih Riwayat Pemesanan
Aktor	Administrator web
Kondisi Awal	Pengguna berada pada halaman Riwayat Pemesanan
Aliran: - Kejadian Normal	1. Sistem menampilkan pilihan pemesanan 2. Pengguna memilih pemesanan yang ingin dilihat dengan menekan lihat. 3. Sistem menampilkan data pemesanan yang dipilih.
- Kejadian Alternatif	-
Kondisi Akhir	Data pemesanan yang dipilih tampil pada sistem.



Gambar 3.9 Diagram Aktivitas Melihat Semua Riwayat Pemesanan

3.4 Spesifikasi Kebutuhan Perangkat Lunak

Bagian ini berisi tentang kebutuhan perangkat lunak. Kebutuhan perangkat lunak dalam sistem ini mencakup kebutuhan fungsional saja. Pada bab ini juga dijelaskan tentang spesifikasi terperinci pada masing-masing kebutuhan fungsional. Rincian spesifikasi dari kasus penggunaan disajikan dalam bentuk tabel.

3.4.1 Kebutuhan Fungsional Sistem

Kebutuhan fungsional berisikan proses-proses yang dibutuhkan dalam sistem dan harus dijalankan. Kebutuhan fungsional sistem dideskripsikan dalam Tabel 3.9.

Tabel 3.9 Kebutuhan Fungsional Sistem

Kode Kebutuhan	Kebutuhan Fungsional	Deskripsi
F-001	Pemesanan Barang	Pengguna dapat melihat barang beserta stok barang dan memasukkan data pemesanan barang.
F-002	Melihat Status Pemesanan	Pengguna dapat melihat status pemesanan yang telah dilakukan dan status pemesanan berubah secara <i>real-time</i> .
F-003	Melihat Riwayat Pemesanan	Pengguna dapat melihat riwayat pemesanan yang telah dilakukan.
F-004	Memperbarui Data Diri	Pengguna dapat memperbarui data diri.
F-005	Manajemen Pemesanan	Pengguna dapat manajemen pemesanan yang masuk.
F-006	Melihat Semua Riwayat Pemesanan	Pengguna dapat melihat semua riwayat pemesanan yang telah dilakukan.
F-007	Manajemen <i>User</i>	Pengguna dapat manajemen data pengguna aplikasi.

3.5 Perancangan Sistem

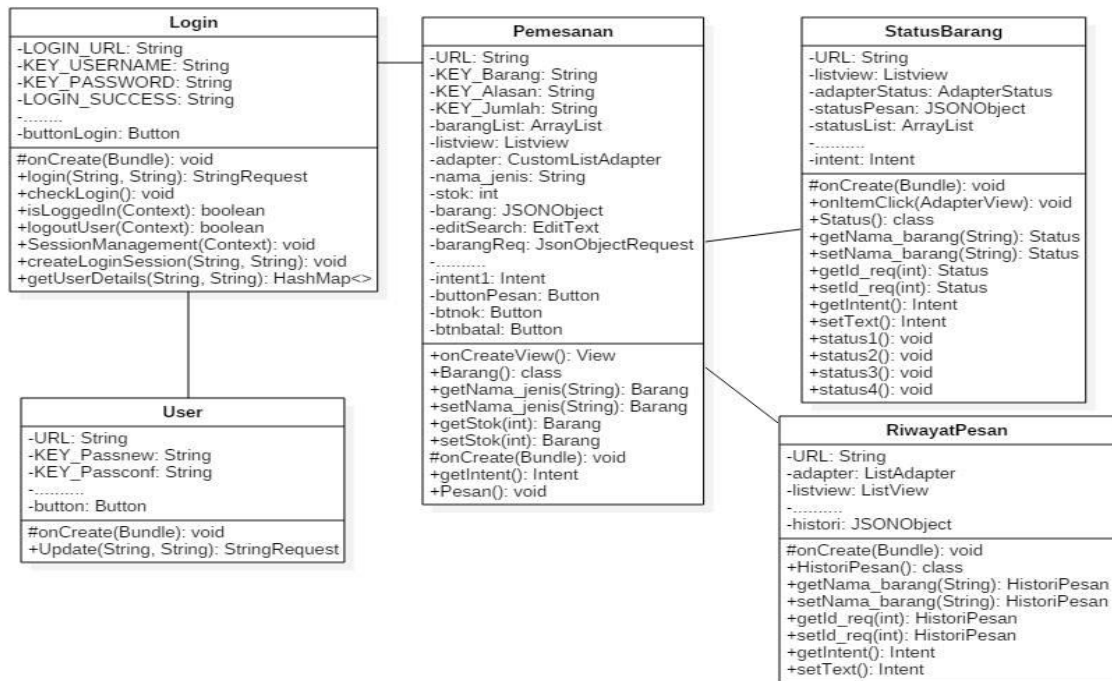
Penjelasan tahap perancangan perangkat lunak dibagi menjadi beberapa bagian yaitu perancangan diagram kelas, perancangan proses analisis, dan perancangan antarmuka.

3.5.1 Perancangan Diagram Kelas

Perancangan diagram kelas berisi kelas-kelas yang dirancang untuk membangun sistem.

3.5.1.1 Diagram Kelas Aplikasi Android

Dalam pembuatan aplikasi pada perangkat bergerak, kelas yang digunakan adalah sebanyak 18 kelas, jika di fungsi kan hanya akan menjadi 4 fungsi utama. Fungsi atau kelas utama tersebut adalah Login, Pemesanan, StatusBarang, RiwayatPesan, dan User. Kelas Login berfungsi sebagai kontroler dari aktivitas login pengguna. Pada kelas ini terdapat 8 operasi dan 7 atribut yang digunakan. Kelas ini berfungsi dari awal pengguna melakukan login, memberikan *session*, dan sampai pengguna melakukan logout. Kelas Pemesanan adalah kelas utama dari aplikasi, dimana kelas ini berfungsi unsebagai kontroller untuk melakukan pemesanan barang. Pada kelas ini terdapat 25 operasi dan 8 atribut yang digunakan. Kelas ini menangani pemilihan barang yang akan dipesan, melakukan pengisian pemesanan, dan memberikan konfirmasi pemesanan sebelum pemesanan benar-benar sesuai dengan yang diharapkan. Kelas StatusBarang berfungsi sebagai kontroler dari aktivitas melihat status pemesanan. Pada kelas ini terdapat 13 operasi dan 11 atribut yang digunakan. Kelas ini menangani perubahan status pemesanan yang telah dilakukan. Kelas Riwayat Pemesanan berfungsi sebagai kontroler dari aktivitas melihat riwayat pemesanan. Pada kelas ini terdapat 9 operasi dan 10 atribut yang digunakan. Kelas ini menangani riwayat pemesanan yang telah dilakukan oleh pengguna. Kelas User berfungsi sebagai kontroler untuk aktivitas memperbarui data diri. Pada kelas ini terdapat 2 operasi dan 9 atribut. Kelas ini menangani perubahan data pengguna jika pengguna ingin memperbarui data diri. Diagram kelas dapat dilihat pada Gambar 3.10.



Gambar 3.10 Diagram Kelas Aplikasi Perangkat Bergerak

3.5.2 Perancangan Algoritma

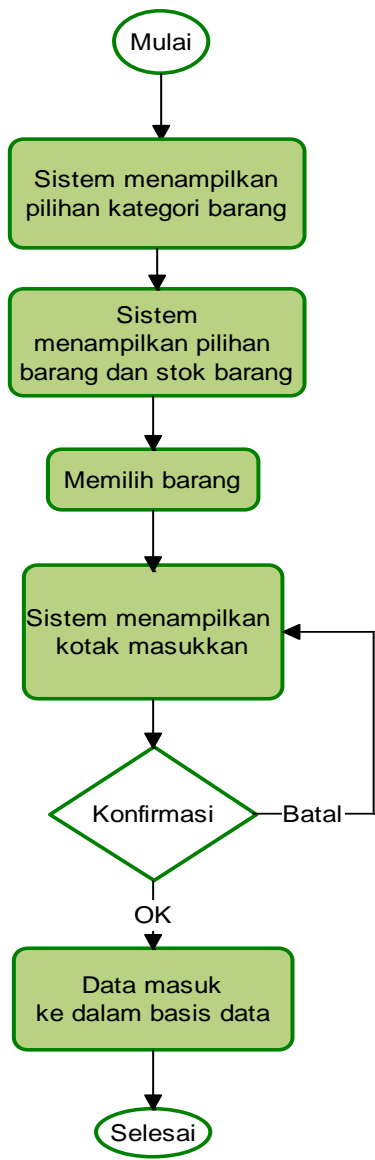
Pada subbab ini akan dijelaskan perancangan algoritma dalam proses yang akan digunakan pada pembuatan aplikasi.

3.5.2.1 Algoritma Memasukkan Data Pemesanan Barang

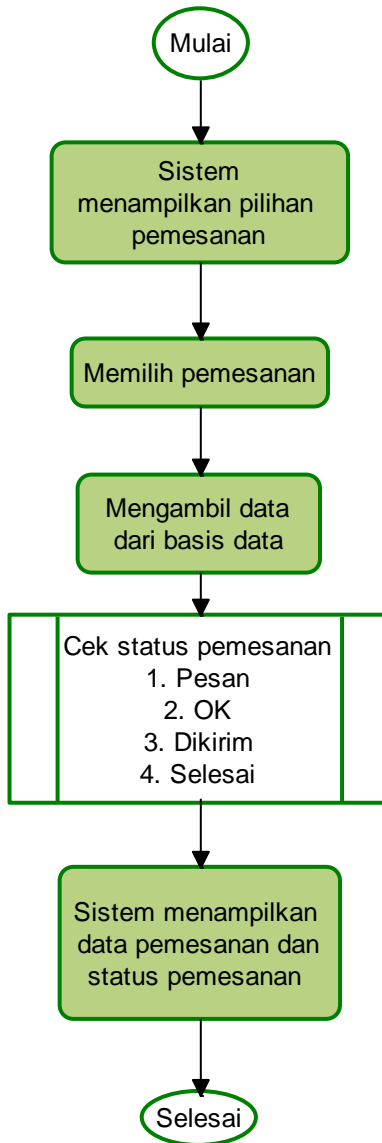
Ini adalah proses awal untuk sistem ini, setelah pengguna memilih menu pesan barang sistem akan menampilkan pilihan kategori barang. Kemudian setelah sistem menerima masukan dari pengguna yaitu pilihan kategori yang dipilih, sistem akan menampilkan pilihan barang beserta stok barang yang tersedia sesuai dengan kategori yang telah dipilih. Pengguna memilih barang yang akan dipesan, sistem akan menampilkan beberapa kotak masukan untuk informasi pemesanan. Pengguna mengisi kotak masukan jumlah, alasan pemesanan, dan alamat pengiriman barang. Pengguna dapat melihat daftar pemesanan di keranjang belanja, pengguna juga dapat menghapus barang yang telah dipilih tadi. Selanjutnya pengguna menekan tombol proses dimana pengguna harus memasukkan informasi pemesanan. Setelah semua terisi pengguna menekan tombol pesan, sistem akan menampilkan verifikasi pemesanan. Diagram alir dapat dilihat pada Gambar 3.11.

3.5.2.2 Algoritma Menampilkan Status Pemesanan

Proses ini dimulai setelah pengguna melakukan pemesanan barang. Sistem akan menampilkan pilihan pemesanan yang telah dilakukan, pengguna memilih salah satu pemesanan jika ingin melihat status pemesanannya. Setelah memilih, sistem akan menampilkan data pemesanan beserta status pemesanan berupa gambar secara *real-time*. Status pemesanan terdiri dari 4 macam yaitu 1 atau pesan, 2 atau ok, 3 atau dikirim, dan 4 atau selesai. Diagram alir dapat dilihat pada Gambar 3.12.



Gambar 3.11 Diagram Alir Algoritma Pemesanan Barang

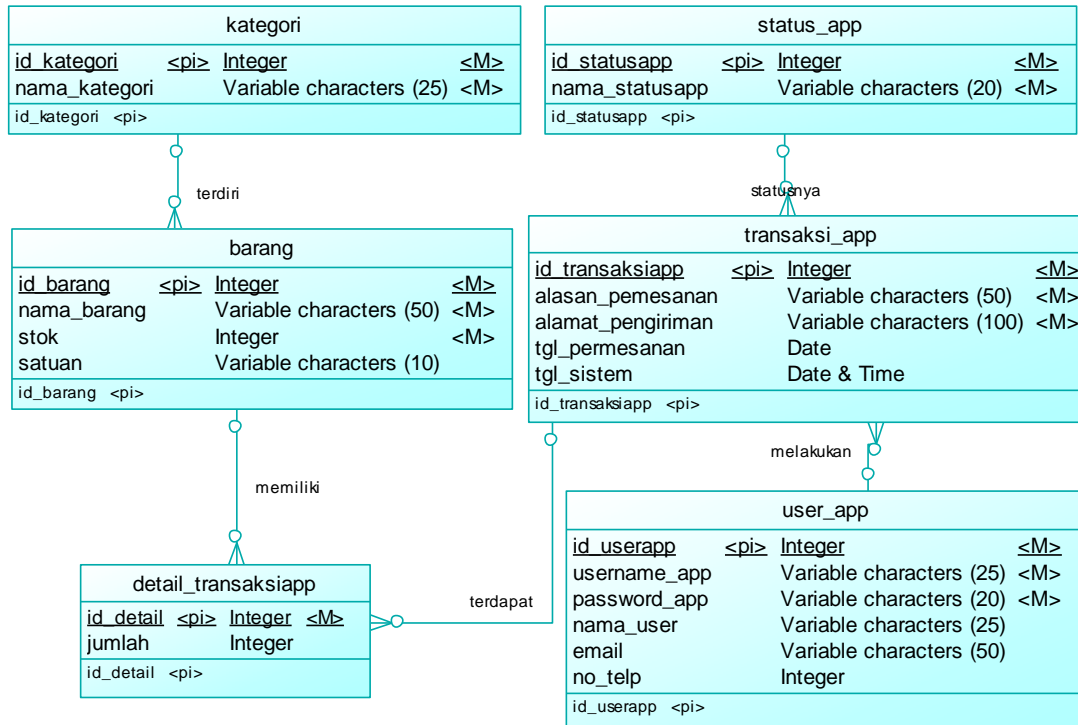


Gambar 3.12 Diagram Alir Algoritma Status Pemesanan

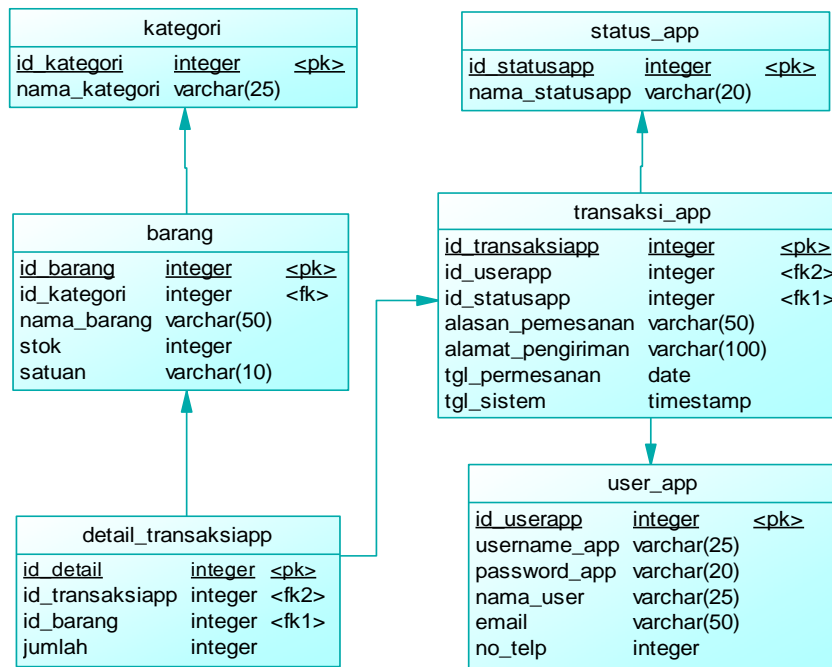
3.5.3 Perancangan Basis Data

Dalam perancangan basis data, sistem akan menggunakan 6 entitas yang direalisasikan dalam bentuk tabel. 6 tabel tersebut adalah `user_app`, `barang`, `kategori`, `transaksi_app`, `detail_transaksiapp`, dan `status_app`. Tabel `user_app` adalah tabel yang berisi data pengguna aplikasi perangkat bergerak. Tabel ini memiliki 6 atribut yaitu `id_userapp` sebagai *primary_key*, `username_app`, `password_app`, `nama_user`, `email`, dan `no_telp`. Tabel `barang` adalah tabel yang berisi data barang-barang yang berada di gudang. Tabel ini memiliki 4 atribut yaitu `id_barang` sebagai *primary_key*, `nama_barang`, `stok`, dan `satuan`. Tabel `kategori` adalah tabel yang berisi data kategori. Tabel ini memiliki 2 atribut yaitu `id_kategori` sebagai *primary_key*, dan `nama_kategori`. Tabel `transaksi_app` adalah tabel utama yang berisi data pemesanan barang. Tabel ini memiliki 4 atribut yaitu `id_transaksiapp` sebagai *primary_key*, `alasan_pemesanan`, `alamat_pengiriman`, dan `tgl_pemesanan`. Tabel `detail_transaksiapp` adalah tabel yang berisi data detail pemesanan barang. Tabel ini memiliki 2 atribut yaitu `id_detail` sebagai *primary_key* dan `jumlah`. Tabel `status_app` adalah tabel yang berisi jenis-jenis status pemesanan. Tabel ini memiliki 2 atribut yaitu `id_statusapp` sebagai *primary_key* dan `nama_statusapp`.

Tabel gudang memiliki relasi terhadap tabel `id_barang` sebagai *foreign_key*. Tabel `user_app` dan `status_app` memiliki relasi terhadap tabel `transaksi_app` sebagai *foreign_key*. Tabel `transaksi_app` dan `barang` memiliki relasi terhadap tabel `detail_transaksi` sebagai *foreign_key*. Dari perancangan ini akan di implementasikan pada basis data MySQL. CDM dari aplikasi ini dapat dilihat pada Gambar 3.13 dan PDM nya dapat dilihat pada Gambar 3.14.



Gambar 3.13 *Conceptual Data Model*

Gambar 3.14 *Physical Data Model*

3.5.4 Perancangan Antarmuka

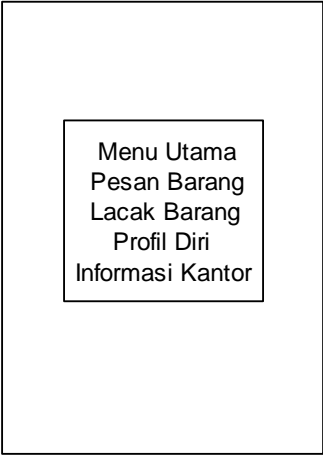
Pada subbab ini akan dijelaskan perancangan antarmuka sistem yang akan dibuat.

3.5.4.1 Antarmuka Aplikasi Perangkat Bergerak

Pada aplikasi perangkat bergerak, antarmuka yang akan ditampilkan meliputi antarmuka halaman login, menu utama, pesan barang, melihat status pemesanan, melihat riwayat pemesanan, informasi kantor, dan memperbarui data diri. Keterangan dan deskripsi perancangan antarmuka dapat dilihat pada Tabel 3.10.

Tabel 3.10 Perancangan Aplikasi Perangkat Bergerak

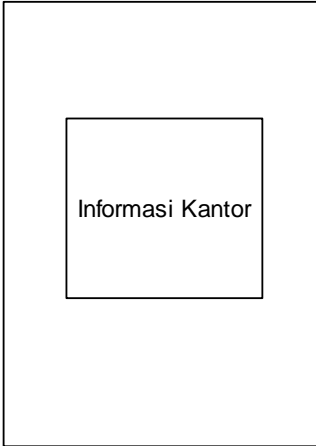
Antarmuka	Keterangan
Login	<p>Deskripsi</p> <p>Pada halaman login ditampilkan dua buah kotak masukan untuk memasukkan <i>username</i> dan <i>password</i> pengguna.</p> <div><div>Username</div><div>Password</div><div>Login</div></div> <p>Gambar 3.15 Perancangan Antarmuka Login</p>

Menu Utama	<p>Deskripsi</p> <p>Pada menu utama sistem menampilkan halaman dengan empat menu yaitu menu pesan barang, lacak barang, profil data diri, dan informasi kantor. Menu utama akan disajikan dengan bentuk gambar.</p> <p>Perancangan</p>  <p>Gambar 3.16 Perancangan Antarmuka Menu Utama</p>
Pesan Barang	<p>Deskripsi</p> <p>Di halaman ini sistem menampilkan barang berupa pilihan kategori barang kemudian pilihan barang yang dapat digeser ke bawah dan ke atas. Setelah itu halaman ini menampilkan nama barang yang dipesan dan stok barang yang</p>

	<p>tersedia. Terdapat 3 kotak masukan untuk memasukkan jumlah barang, alasan pemesanan, dan alamat pengiriman barang. Pengguna dapat menekan tombol pesan untuk memasukkan data yang ada pada kotak isian ke basis data.</p> <p>Perancangan</p> <div><div>Nama Barang Stok Barang</div><div>Jumlah Pemesanan</div><div>Alasan Pemesanan</div><div>Alamat Pengiriman</div><div>Pesan</div></div> <p>Gambar 3.17 Perancangan Antarmuka Pesan Barang</p>
<p>Melihat Status Pemesanan</p>	<p>Deskripsi</p> <p>Sistem menampilkan data status pemesanan yang telah dilakukan oleh pengguna. Status pemesanan ini ditampilkan setelah pengguna memilih salah satu pemesanan. Terdapat 6 atribut</p>

	<p>yang akan ditampilkan pada halaman ini yaitu nomor pemesanan, tgl pemesanan, nama barang, jumlah, alasan pemesanan, dan alamat pengiriman. Dan satu atribut akan ditampilkan dengan bantuan gambar yaitu status pemesanan.</p> <p>Perancangan</p> <div data-bbox="539 494 875 967"> <p>Nomor Pemesanan Tgl Pemesanan</p> <p>Nama Barang : Jumlah : Alasan Pemesanan : Alamat Pengiriman :</p> <div data-bbox="589 807 827 876"> <p>Status Pemesanan</p> </div> </div> <p>Gambar 3.18 Perancangan Antarmuka Melihat Status Pemesanan</p>
<p>Melihat Riwayat Pemesanan</p>	<p>Deskripsi</p> <p>Sistem menampilkan data riwayat pemesanan yang telah dilakukan oleh pengguna. Data riwayat pemesanan ini ditampilkan setelah pengguna memilih salah satu riwayat pemesanan yang ingin dilihat oleh pengguna. Terdapat 6</p>

	<p>atribut yang akan ditampilkan pada halaman ini yaitu nomor pemesanan, tgl pemesanan, nama barang, jumlah, alasan pemesanan, dan alamat pengiriman</p> <p>Perancangan</p> <div data-bbox="507 427 844 900" style="border: 1px solid black; padding: 10px; margin: 10px auto; width: fit-content;"> <p>Nomor Pemesanan Tgl Pemesanan</p> <p>Nama Barang : Jumlah : Alasan Pemesanan : Alamat Pengiriman :</p> </div> <p>Gambar 3.19 Perancangan Antarmuka Melihat Riwayat Pemesanan</p>
<p>Informasi Kantor</p>	<p>Deskripsi</p> <p>Halaman ini akan menampilkan informasi kantor. Informasi kantor yaitu alamat kantor, nomor telepon kantor, dan jam buka kantor. Data pada halaman ini bersifat dinamis karena data tidak akan berubah.</p>

	<p>Perancangan</p>  <p>Gambar 3.20 Perancangan Antarmuka Informasi Kantor</p>
Profil Diri	<p>Deskripsi</p> <p>Dihalaman ini terdapat 4 masukan untuk memasukkan <i>username</i>, nomor telepon, <i>password</i> baru, dan <i>confirmation password</i>. Tombol simpan digunakan untuk menyimpan data yang telah dimasukkan ke dalam basis data. Pada halaman ini data yang sebelumnya akan ditampilkan jika pengguna ingin memperbarui beberapa atribut, maka yang diisi masukan yang ingin diperbarui.</p>

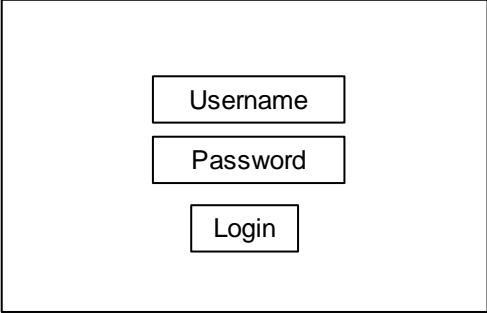
	<p>Perancangan</p> <div style="border: 1px solid black; padding: 10px; margin: 10px auto; width: 80%;"> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; text-align: center;">Username</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; text-align: center;">No Telp</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; text-align: center;">Password</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; text-align: center;">Konfirmasi Password</div> <div style="border: 1px solid black; padding: 5px; margin-bottom: 5px; text-align: center;">Simpan</div> </div> <p style="text-align: center;">Gambar 3.21 Perancangan Antarmuka Profil Diri</p>
--	---

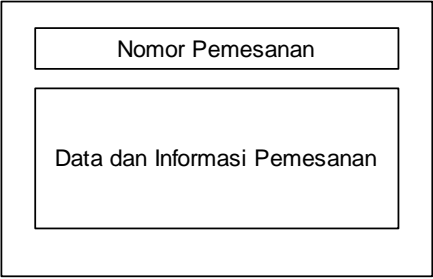
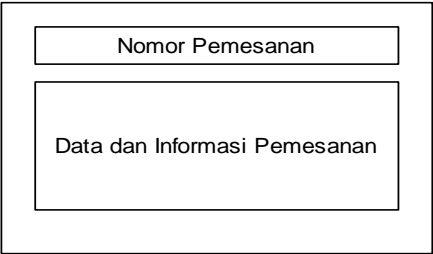
3.5.4.2 Antarmuka Aplikasi Web

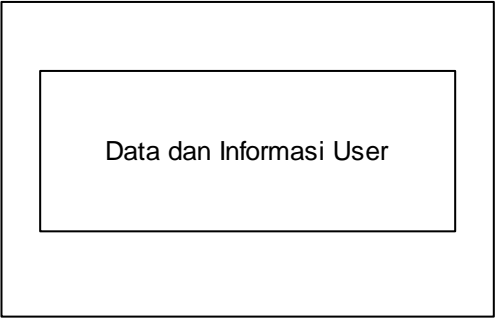
Dalam perancangan antarmuka aplikasi web terdapat 4 halaman yang digunakan. Halaman yang akan digunakan adalah halaman login, manajemen status pemesanan, rekap pemesanan, dan manajemen data user. Daftar dan keterangan perancangan antarmuka dapat dilihat pada Tabel 3.11.

Tabel 3.11 Perancangan Antarmuka Aplikasi Web

Antarmuka	Keterangan
Login	<p>Deskripsi</p> <p>Pada halaman login ditampilkan dua buah</p>

	<p>kotak masukan untuk memasukkan <i>username</i> dan <i>password</i> pengguna.</p> <p>Perancangan</p>  <p>Gambar 3.22 Perancangan Antarmuka Login Web</p>
<p>Manajemen Pemesanan</p>	<p>Deskripsi</p> <p>Halaman ini menampilkan data pemesanan yang telah dilakukan dengan memilih salah satu pemesanan dan memanajemen pemesanan yang telah dipilih. Terdapat 2 tombol disetiap pemesanan yaitu tombol edit dan lihat. Halaman ini juga terdapat pilihan <i>dropdown</i> jika pengguna ingin menampilkan data pemesanan berdasarkan status pemesanannya.</p>

	<p>Perancangan</p>  <p>Gambar 3.23 Perancangan Antarmuka Manajemen Pemesanan</p>
<p>Seluruh Riwayat Pemesanan</p>	<p>Deskripsi</p> <p>Halaman ini menampilkan riwayat pemesanan yang telah dilakukan oleh pengguna tersebut. Dengan memilih salah satu pemesanan maka detail pemesanan akan ditampilkan.</p> <p>Perancangan</p>  <p>Gambar 3.24 Perancangan Antarmuka Seluruh Riwayat Pemesanan</p>

Manajemen User	Data	Deskripsi <p>Halaman ini menampilkan kotak isian untuk mengisi data user. Setelah mengisi data user, pengguna dapat menekan tombol simpan untuk memasukkan data yang ada pada kotak isian ke basis data.</p> Perancangan  <p>Gambar 3.25 Perancangan Antarmuka Manajemen Data User</p>
-----------------------	-------------	---

[Halaman ini sengaja dikosongkan]

BAB IV IMPLEMENTASI

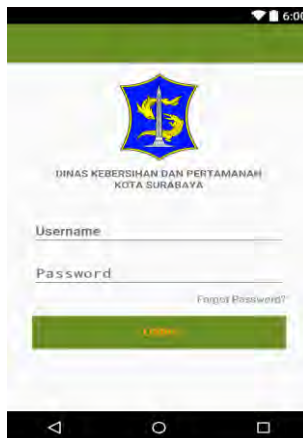
Pada bab ini akan dibahas mengenai implementasi yang akan dijelaskan dengan bentuk antarmuka dan halaman sesuai dengan analisis dan perancangan perangkat lunak yang telah dibuat.

4.1 Implementasi Antarmuka

Pada subbab ini akan dijelaskan tentang implementasi antarmuka sistem yang menjadi bagian terluar sekaligus bagian yang akan berinteraksi langsung dengan pengguna.

4.1.1 Antarmuka Aktivitas Login

Pada antarmuka login pengguna, terdapat dua kotak masukan untuk memasukkan *username* dan *password* pengguna. Implementasi dapat dilihat pada Gambar 4.1.



Gambar 4.1 Antarmuka Login

4.1.2 Antarmuka Menu Utama

Pada antarmuka menu utama terdapat 3 menu yaitu, pemesanan barang, lacak barang, dan informasi kantor. Tampilan antarmuka ini dapat dilihat pada Gambar 4.2.



Gambar 4.2 Antarmuka Menu Utama

4.1.3 Antarmuka Pemesanan Barang

Sebelum masuk pada antarmuka pemesanan barang, pengguna menekan tombol pesan barang di halaman utama. Sistem akan menampilkan halaman antarmuka pemesanan barang. Pada antarmuka pemesanan barang di halaman awal terdapat pilihan kategori barang. Pilihan kategori barang berupa daftar dan dapat di geser ke atas maupun ke bawah. Setelah itu pengguna dapat memilih kategori barang yang ingin di pesan. Tampilan antarmuka dapat dilihat pada Gambar 4.3.



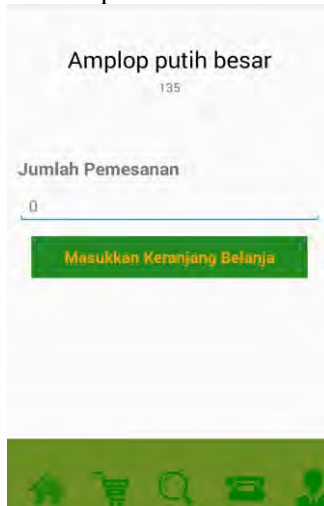
Gambar 4.3 Antarmuka Pilih Kategori Barang

Setelah pengguna memilih kategori barang, sistem akan menampilkan halaman jenis barang berdasarkan kategori. Sistem menampilkan daftar jenis barang dan dapat di geser ke atas dan ke bawah. Tampilan dapat dilihat pada Gambar 4.4 Pengguna memilih barang yang ingin dipesan.



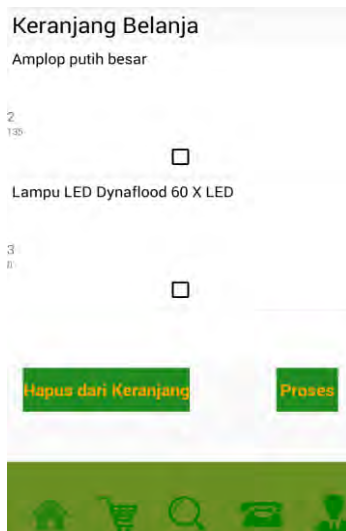
Gambar 4.4 Antarmuka Pemesanan Barang

Setelah pengguna memilih barang yang ingin dipesan aplikasi akan melanjutkan ke halaman pemesanan selanjutnya. Di halaman selanjutnya terdapat satu kotak masukan untuk memasukkan jumlah barang yang ingin dipesan. Tampilan antarmuka ini dapat dilihat pada Gambar 4.5.

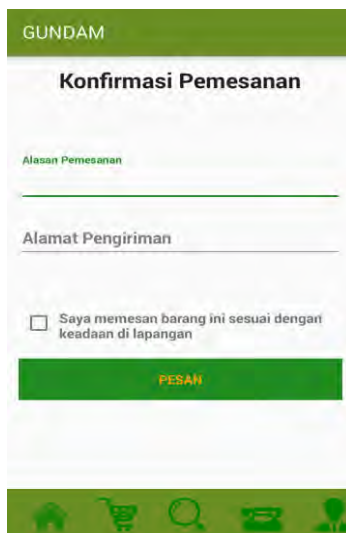


Gambar 4.5 Antarmuka Memasukkan Jumlah Pemesanan

Setelah selesai memilih barang halaman aplikasi akan melanjutkan ke halaman pemesanan selanjutnya dimana sistem akan menampilkan daftar keranjang belanja pemesanan. Gambar dapat dilihat pada Gambar 4.6. Jika pengguna ingin menghapus barang yang telah dipilih, terdapat tombol centang dan tombol hapus barang. Pengguna mengisi tombol centang di barang yang ingin dihapus kemudian menekan tombol hapus barang. Kemudian pengguna menekan tombol proses jika dirasa daftar barang sudah sesuai dengan kebutuhan. Langkah selanjutnya terdapat 2 kotak masukan yaitu alasan pemesanan dan alamat pengiriman. Terdapat satu tombol yang berfungsi untuk memasukkan data kedalam basis data yaitu tombol pesan. Tampilan antarmuka ini dapat dilihat pada Gambar 4.7.



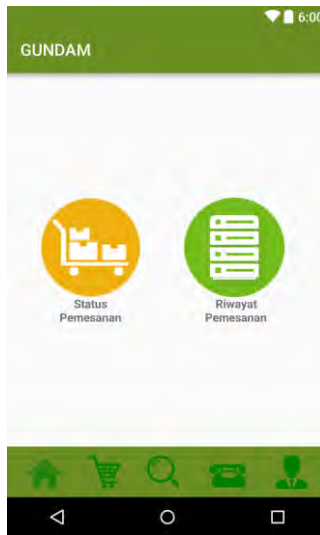
Gambar 4.6 Antarmuka Keranjang Belanja



Gambar 4.7 Antarmuka Konfirmasi Pemesanan

4.1.4 Antarmuka Lacak Pemesanan

Pada antarmuka menu lacak pemesanan, terdapat dua tombol yaitu tombol status pemesanan yang berfungsi untuk melanjutkan ke halaman status pemesanan dan tombol riwayat pemesanan yang berfungsi untuk melanjutkan ke halaman riwayat pemesanan. Tampilan antarmuka dapat dilihat pada Gambar 4.8.



Gambar 4.8 Antarmuka Lacak Pemesanan

4.1.5 Antarmuka Status Pemesanan

Pada antarmuka status pemesanan terdapat pilihan pemesanan yang telah dibuat. Tampilan dapat dilihat pada Gambar 4.9. Setelah pengguna memilih pemesanan aplikasi akan melanjutkan ke halaman status pemesanan selanjutnya. Di halaman ini terdapat informasi pemesanan yang telah dilakukan beserta status pemesanan berupa gambar secara *real-time*. Tampilan dapat dilihat pada Gambar 4.10.



Gambar 4.9 Antarmuka Melihat Status Pemesanan



Gambar 4.10 Antarmuka Detail Status Pemesanan

4.1.6 Antarmuka Riwayat Pemesanan

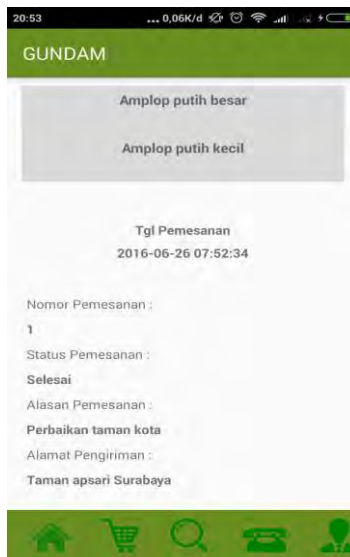
Pada antarmuka riwayat pemesanan terdapat pilihan pemesanan yang telah dilakukan. Tampilan dapat dilihat pada Gambar 4.11. Setelah pengguna memilih riwayat pemesanan yang ingin dilihat, aplikasi akan melanjutkan ke halaman status riwayat pemesanan selanjutnya. Di halaman ini terdapat informasi pemesanan yang telah dilakukan. Tampilan dapat dilihat pada Gambar 4.12.



Gambar 4.11 Antarmuka Melihat Riwayat Pemesanan

4.1.1 Antarmuka Informasi Kantor

Pada antarmuka informasi kantor, menampilkan lokasi kantor berada beserta nomor telepon yang dapat dihubungi dan jam buka dan tutup kantor. Tampilan dapat dilihat pada Gambar 4.13.



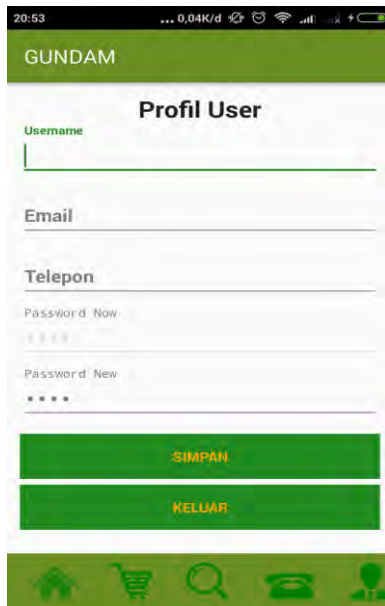
Gambar 4.12 Antarmuka Detail Riwayat Pemesanan



Gambar 4.13 Antarmuka Informasi Kantor

4.1.1 Antarmuka Menu Profil Diri

Pada antarmuka menu user, terdapat lima kotak masukan yang berfungsi untuk memperbarui data user. Kotak masukan pertama yaitu untuk memperbarui *username* baru, kotak kedua untuk memperbarui nomor telepon, kotak ketiga, keempat, dan kelima untuk memperbarui *password*. Tampilan antarmuka dapat dilihat pada Gambar 4.14.



Gambar 4.14 Antarmuka Profil Diri





4.1.2 Antarmuka Halaman Login Web

Pada halaman login terdapat dua kotak masukan untuk memasukkan *username* dan *password* agar dapat melakukan login. Antarmuka halaman ini dapat dilihat pada Gambar 4.15.

Gambar 4.15 Antarmuka Halaman Login Web

4.1.3 Antarmuka Halaman Status Pemesanan

Pada halaman status pemesanan pengguna dapat melihat pemesanan baru yang telah dilakukan. Pada antarmuka tersebut terdapat informasi data pemesanan. Terdapat tombol yang digunakan pengguna yang berfungsi untuk mengubah status pemesanan beserta memperbarui data pemesanan. Antarmuka halaman ini dapat dilihat pada Gambar 4.16. Pengguna dapat mengubah data pemesanan beserta status pemesanannya, dapat dilihat pada Gambar 4.17.

NO	NO PEMESANAN	NAMA PENGGUNA	NO TELP	ALAMAT PENGIRIMAN	TGL PEMESANAN	STATUS	AKSI
1	1	sho@i	812347266	Jalan Ahmad Yani	2016-06-11 03:18:49	Pengiriman	 
2	2	sho@i	812347266	Menur Plumpungan	2016-06-20 00:11:16	Selesai	 

Gambar 4.16 Antarmuka Status Pemesanan

Edit Status

Status Pemesanan
Pemesanan
Edit Status

Status

Pengiriman

ID Pemesanan

1

Nama Pemesan

shoffi

No Telp Pemesan

812347266

Alasan Pemesanan

Perbaikan

Alamat Pengiriman

Jalan Ahmad Yani

Status Pemesanan

Pengiriman

Tgl Pemesanan

2016-06-11 03:18:49

Cari:

NO	NAMA BARANG	JUMLAH PEMESANAN
1	Amplop putih kecil	10
2	Lampu LED Dynaflood 36 X LED	2

NO	NAMA BARANG	JUMLAH PEMESANAN
----	-------------	------------------

Gambar 4.17 Antarmuka Edit Status Pemesanan

Detail Riwayat Pemesanan

Riwayat Pemesanan
Pemesanan
Detail Riwayat

ID Pemesanan
2

Nama Pemesan
shoffi

Alasan Pemesanan
Restock

Status Pemesanan
Selesai

No Telp Pemesan
812347266

Alamat Pengiriman
Menur Plumpungan

Tgl Pemesanan
2016-06-20 00:11:16

Cari:

NO	NAMA BARANG	JUMLAH PEMESANAN
1	Amplop putih besar	10
2	Lampu LED Dynaflood 60 X LED	20

Menampilkan 1 sampai 2 entri dari 2 total entri

Sebelumnya
1
Berikutnya

Gambar 4.18 Antarmuka Detail Riwayat Pemesanan

4.1.4 Antarmuka Halaman Riwayat Pemesanan

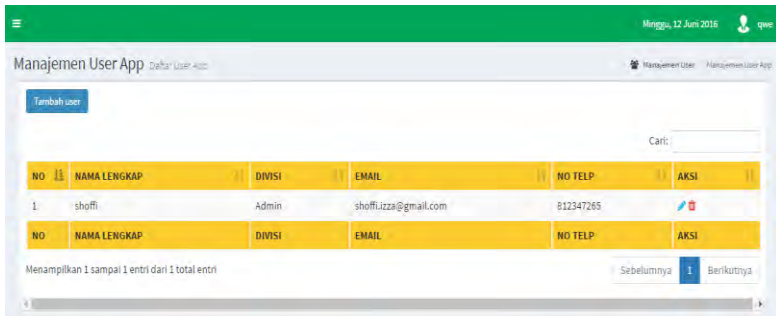
Pada halaman riwayat pemesanan pengguna dapat melihat riwayat pemesanan yang telah dilakukan. Antarmuka halaman ini dapat dilihat pada Gambar 4.19. Terdapat satu kotak masukan yang berfungsi untuk mencari riwayat pemesanan yang diinginkan dan tombol yang digunakan pengguna untuk melihat informasi suatu pemesanan tertentu. Antarmuka lihat detail pemesanan dapat dilihat pada Gambar 4.18.

NO	NO PEMESANAN	NAMA PENGGUNA	NO TELP	ALAMAT PENGIRIMAN	TGL PEMESANAN	STATUS	AKSI
1	2	shoffi	812347266	Menur Plumpungan	2016-06-20 00:11:16	Selesai	

Gambar 4.19 Antarmuka Halaman Rekap Pemesanan

4.1.5 Antarmuka Halaman Manajemen User

Pada halaman manajemen user pengguna dapat melihat pengguna yang menggunakan aplikasi. Terdapat tiga tombol yang berfungsi untuk menambah pengguna, melihat pengguna, dan menghapus pengguna. Antarmuka halaman ini dapat dilihat pada Gambar 4.20. Pengguna juga dapat menambah user. Terdapat 6 kotak isian pada halaman tambah user, dapat dilihat pada Gambar 4.21. Selain itu pengguna juga dapat mengubah data user, dapat dilihat pada Gambar 4.22.



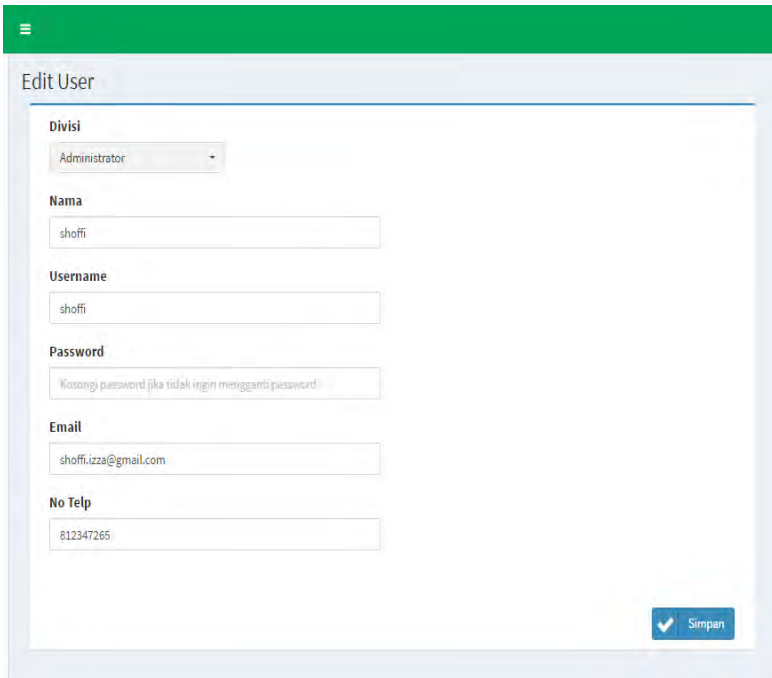
Gambar 4.20 Antarmuka Halaman Manajemen User

The screenshot shows the 'Tambah User Baru' form. It has a green header bar with a menu icon. The title 'Tambah User Baru' is displayed. The form contains the following fields:

- Divisi:** A dropdown menu with the text '--Pilih Divisi--'.
- Nama:** A text input field.
- Username:** A text input field.
- Password:** A text input field.
- Ulangi Password:** A text input field.
- Email:** A text input field.
- No Telp:** A text input field.

At the bottom right, there is a blue button with a checkmark icon and the text 'Tambah'.

Gambar 4.21 Antarmuka Halaman Tambah User



Edit User

Divisi
Administrator

Nama
shoffi

Username
shoffi

Password
Kosongi password jika tidak ingin mengganti password

Email
shoffi.izza@gmail.com

No Telp
812347265

✓ Simpan

Gambar 4.22 Antarmuka Edit User

4.2 Implementasi Fitur

Pada subbab ini akan dijelaskan implementasi yang berupa baris kode yang sesuai dengan kebutuhan fungsional.

4.2.1 Fitur Pemesanan Barang

Fitur ini diimplementasikan pada aplikasi Android, secara fungsional fitur ini dilakukan untuk memesan barang. Pada aplikasi pengguna dapat memesan barang dengan jumlah barang dan alasan pemesanan. Pengguna dapat melihat pilihan barang dan stok barang secara *real-time*.

```

1.     private void pesanTransaksi() {
2.         final String getAlasan = alasan.getText().toString().
           trim();
3.         final String getAlamat = alamat.getText().toString().
           trim();
4.
5.         StringRequest stringRequest = new StringRequest(Request.M
           ethod.POST, url,
6.             new Response.Listener<String>() {
7.                 @Override
8.                 public void onResponse(String response) {
9.                     try {
10.                        JSONObject transaksi = new JSONObject(response);
11.
12.                        id_transaksi = transaksi.getInt("id_t
ransaksiapp");
13.
14.                        for (int i = 0; i < mCartList.size();
                           i++) {
15.                            mCartList.get(i).selected = false;
16.
17.                            detailTransaksi(mCartList.get(i).g
                               etId_Jenis(), mCartList.get(i).getJumlah(), id_transaksi);
18.                        }
19.                    } catch (JSONException e) {
20.                        e.printStackTrace();
21.                    }
22.                }, new Response.ErrorListener() {
23.                    @Override
24.                    public void onErrorResponse(VolleyError error) {
25.
26.                    }
27.                }) {
28.                    protected Map<String, String> getParams() {
29.                        Map<String, String> params = new HashMap<String, Stri
ng>();
30.                        params.put(KEY_Alasan, getAlasan);
31.                        params.put(KEY_Alamat, getAlamat);
32.
33.                        return params;
34.                    }
35.                };

```

```

35. RequestQueue requestQueue = Volley.newRequestQueue(Confirmati
    on.this);
36. requestQueue.add(stringRequest);

```

Kode Sumber 4.1 Fitur Pemesanan Barang

4.2.2 Fitur Melihat Status Pemesanan

Fitur ini di implementasikan pada aplikasi perangkat bergerak, secara fungsional fitur ini digunakan untuk melihat status pemesanan yang telah dilakukan.

```

1. String statuspemesanan = intent.getStringExtra("status");
2.     if (statuspemesanan.equals("1")) {
3.         status1();
4.     }
5.     else if (statuspemesanan.equals("2")){
6.         status2();
7.     }else if (statuspemesanan.equals("3")){
8.         status3();
9.     } else {
10.        status4();
11.    }

```

Kode Sumber 4.2 Fitur Melihat Status Pemesanan

4.2.3 Fitur Melihat Riwayat Pemesanan

Fitur ini diimplementasikan pada aplikasi perangkat bergerak, secara fungsional fitur ini digunakan untuk melihat histori pemesanan yang telah dilakukan. Pada aplikasi pengguna dapat melihat histori pemesanan dan informasi-informasi yang dibutuhkan.

```

1. JsonObjectRequest statusReq = new JsonObjectRequest(Request.M
    ethod.GET,
2.                url, null, new Response.Listener<JSONObject>(
    ) {

```

```

3.
4.         @Override
5.         public void onResponse(JSONObject response) {
6.             JSONArray content = null;
7.             Log.d(String.valueOf(content), response.toString());
8.             try {
9.                 content = response.getJSONArray("content");
10.                Log.d("panjang", String.valueOf(content.length()));
11.                for (int i = 0; i < content.length(); i++) {
12.
13.                    JSONObject statusB = (JSONObject) content.get(i);
14.
15.                    HistoriPesan detailH = new HistoriPesan();
16.                    detailH.setNama_barang(nama_barang);
17.
18.                    Log.d("nama_jenis", nama_barang);
19.                    detailList.add(detailH);
20.
21.                }
22.
23.            } catch (JSONException error) {
24.                error.printStackTrace();
25.            }
26.            adapterHistori.notifyDataSetChanged();
27.        }
28.    }, new Response.ErrorListener() {
29.        @Override
30.        public void onErrorResponse(VolleyError error) {
31.
32.        }
33.    });

```

Kode Sumber 4.3 Fitur Melihat Riwayat Pemesanan

4.2.4 Fitur Memperbarui Data User

Fitur ini diimplementasikan pada aplikasi Android, secara fungsional fitur ini digunakan untuk memperbarui data pengguna.

```

1.  public void Update(){
2.
3.      final String Username = mUsername.getText().toString().trim();
4.      final String Password = mPassnow.getText().toString().trim();
5.      final String Passnew = mPassnew.getText().toString().trim();
6.      final String Telp = mTelp.getText().toString().trim();
7.      final String Email = mEmail.getText().toString().trim();
8.
9.      StringRequest stringRequest = new StringRequest(Request.Method.POST, URL,
10.
11.                  new Response.Listener<String>() {
12.
13.                      @Override
14.                      public void onResponse(String response) {
15.
16.                          Toast.makeText(User.this, response, Toast.LENGTH_LONG).show();
17.                          Intent sukses = new Intent(getApplicationContext(), User.class);
18.                          startActivity(sukses);
19.                      }
20.                  }, new Response.ErrorListener() {
21.
22.                      @Override
23.                      public void onErrorResponse(VolleyError or error) {
24.
25.                          Toast.makeText(User.this, error.toString(), Toast.LENGTH_LONG).show();
26.                      }
27.                  }) {
28.
29.                  protected Map<String, String> getParams() {

```



```

25.
26. Map<String, String> params = new
    HashMap<String, String>();
27. params.put(KEY_Username, Username
    );
28. params.put(KEY_Password, Password
    );
29. params.put(KEY_Passnew, Passnew);
30. params.put(String.valueOf(KEY_Tel
    p), Telp);
31. params.put(KEY_Email, Email);
32.
33. return params;
34. }
35. };
36. RequestQueue requestQueue = Volley.newRequestQueue(User.this)
    ;
37. requestQueue.add(stringRequest);

```

Kode Sumber 4.4 Fitur Memperbarui Data User

4.2.5 Fitur Manajemen Pemesanan

Fitur ini diimplementasikan pada aplikasi web, secara fungsional fitur ini digunakan untuk manajemen pemesanan. Pengguna dapat melihat pemesanan yang telah dilakukan dan dapat merubah dan memperbarui data pemesanan beserta status pemesanan.

```

1. public function updateStatus($id, $data){
2.     $result = $this->db-
    >get_where('transaksi_app', array('id_transaksiapp' =>
        $id));
3.     if ($result->num_rows() < 0)
4.     {
5.         return FALSE;
6.     }
7.     else
8.     {

```

```

9.         $this->db->where('id_transaksiapp', $id);
10.        $this->db->update('transaksi_app', $data);
11.
12.        return TRUE;
13.    }

```

Kode Sumber 4.5 Fitur Manajemen Status Pemesanan

4.2.6 Fitur Melihat Semua Riwayat Pemesanan

Fitur ini diimplementasikan pada aplikasi web, secara fungsional fitur ini digunakan untuk Melihat semua riwayat pemesanan pemesanan. Pengguna dapat melihat pemesanan yang telah dilakukan dan dapat merubah dan memperbarui data pemesanan beserta status pemesanan.

```

1. public function getrekapReq($id){
2.
3.     $q = "SELECT * FROM detail_transaksiapp, transaksi_app, jenis_barang, status_app, user_app
4.         WHERE transaksi_app.id_transaksiapp = detail_transaksiapp.id_transaksiapp
5.         AND jenis_barang.id_jenis = detail_transaksiapp.id_jenis
6.         AND transaksi_app.id_userapp = user_app.id_userapp AND status_app.id_statusapp = transaksi_app.id_statusapp
7.         AND transaksi_app.id_transaksiapp = '$id'";
8.
9.     $query = $this->db->query($q);
10.
11.     if ($query->num_rows() > 0)
12.     {
13.         return $query->result_array();
14.     }
15.     else
16.     {
17.         return FALSE;
18.     }
19.

```

```

20.     }
21.
22.     }

```

Kode Sumber 4.6 Fitur Melihat Semua Riwayat Pemesanan

4.2.7 Fitur Manajemen User

Fitur ini diimplementasikan pada aplikasi web. Secara fungsional fitur ini digunakan untuk manajemen pengguna aplikasi Android. Pengguna dapat menambahkan dan menghapus data pengguna aplikasi Android.

```

1.  public function tambahUser($username, $data){
2.      $result = $this->db->get_where('user_app',
3.          array
4.          (
5.              'username_app' => $username
6.          )
7.      );
8.
9.      if ($result->num_rows() > 0)
10.     {
11.         return FALSE;
12.     }
13.     else
14.     {
15.         $this->db->insert('user_app', $data);
16.         return TRUE;
17.     }
18. }
19.
20.
21. public function update($id, $username, $data){
22.     $q = $this->db-
23.     >get_where('user_app', array('id_userapp' => $id));
24.     $user = $q->row()->username_app;
25.     $result = $this->db-
26.     >get_where('user_app', array('username_app' => $username));

```

```
26.         if ($result->num_rows() > 0){
27.             if ($username != $user){
28.                 return FALSE;
29.             }
30.         }
31.         $this->db->where('id_userapp', $id);
32.         $this->db->update('user_app', $data);
33.         return TRUE;
34.
35.     }
36.
37.
38.     public function hapusUser($id)
39.     {
40.         $this->db->
41.         >delete('user_app', array('id_userapp' => $id));
42.     }
```

Kode Sumber 4.7 Fitur Manajemen User

BAB V

UJI COBA DAN EVALUASI

Pada bab ini akan dijelaskan uji coba yang dilakukan pada aplikasi yang telah dikerjakan serta analisa dari uji coba yang telah dilakukan. Pembahasan pengujian meliputi lingkungan uji coba, skenario uji coba yang meliputi uji kebenaran dan uji kinerja serta analisa setiap pengujian.

5.1 Lingkungan Uji Coba

Lingkungan uji coba meliputi perangkat keras dan perangkat lunak yang dijelaskan sebagai berikut:

Tabel 5.1 Lingkungan Uji Coba

Aplikasi	Perangkat Bergerak	Web
Jenis Perangkat	Perangkat Bergerak	Komputer
Prosesor	Qualcomm Snapdragon S4 Plus MSM8227 CPU Dual-core 1 GHz Krait GPU Adreno 305	Intel Core i3 M330 @ 2,13GHz
Memori	1 GB	2 GB
Sistem Operasi	Android	Windows
Jenis Sistem Operasi	4.3 (Jelly Bean)	Windows 8.1 (x64)

5.2 Dataset Uji Coba

Pada tugas akhir ini, penulis menggunakan dataset dari dinas kebersihan kota surabaya. Data set yang digunakan adalah nama barang dan jumlah barang yang ada digudang pada saat ini. Data set ini digunakan untuk proses utama yaitu pemesanan barang. Terdapat 4 kategori barang dan 151 barang dan masing-masing barang memiliki stok barang. Dilakukan proses uji coba dataset dengan menampilkan semua dataset pada aplikasi perangkat bergerak. Hasil dari uji coba dataset, pada aplikasi perangkat bergerak ditemukan 151 barang beserta stok barangnya.

5.3 Skenario Pengujian

Pada bagian ini akan dijelaskan tentang skenario pengujian yang dilakukan. Pengujian yang dilakukan adalah pengujian kebutuhan fungsionalitas. Pengujian fungsionalitas menggunakan metode kotak hitam (*black box*). Metode ini menekankan pada kesesuaian hasil keluaran sistem.

Pengujian dilakukan untuk mengetahui hasil dari masukan data dan keluaran data. Pada aplikasi Android, masukan data yang dibutuhkan data pemesanan. Keluaran yang diharapkan data pemesanan dapat tampil pada aplikasi dan dapat dilihat status pemesanannya dan dapat melihat riwayat pemesanan yang telah dilakukan. Pada aplikasi web, masukan data yang dibutuhkan adalah masukan manajemen pengguna aplikasi, manajemen data pemesanan, dan Melihat semua riwayat pemesanan. Keluaran yang diharapkan adalah data pemesanan dan semua riwayat pemesanan barang.

Pengujian pertukaran data dan performa sistem juga akan dilakukan. Setelah itu pengujian juga dilakukan kepada responden yang memiliki peran yang sama atau hampir sama dengan pengguna sebenarnya.

5.3.1 Pengujian Fungsionalitas

Pada subbab ini akan dijelaskan tentang pengujian aplikasi dari segi fungsionalitasnya.

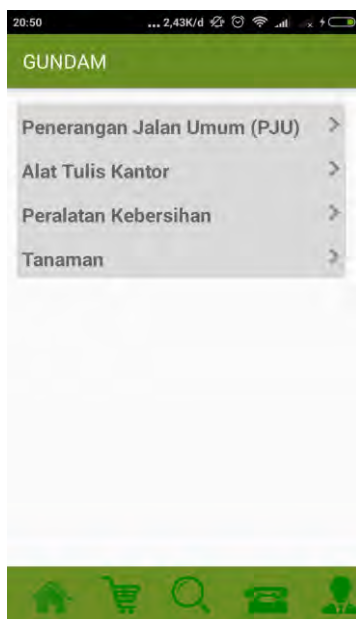
5.3.1.1 Pengujian Fitur Pemesanan Barang

Pada pengujian ini pengguna akan melakukan pemesanan barang. Skenario pengujian pemesanan barang dapat dilihat pada Tabel 5.2 dan hasil pengujian dari memilih kategori barang dapat dilihat pada Gambar 5.1. Hasil pengujian dari memilih barang dapat dilihat pada Gambar 5.2. Hasil pengujian dari memasukkan jumlah barang dapat dilihat pada Gambar 5.3. Hasil pengujian memasukkan kedalam keranjang belanja dapat dilihat pada Gambar 5.4. Hasil pengujian tombol keranjang belanja dapat dilihat pada Gambar 5.5. Hasil pengujian menghapus barang yang ada di keranjang belanja dapat dilihat pada Gambar 5.6. Dan yang terakhir hasil pengujian dari memilih tombol pesan dapat dilihat pada Gambar 5.7.

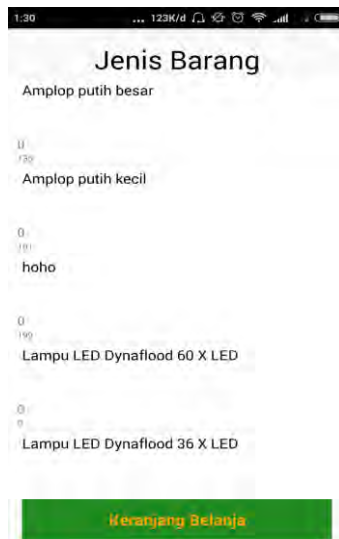
Tabel 5.2 Pengujian Fitur Pemesanan Barang

Test ID	TUC-001
Kasus Penggunaan	Pemesanan Barang
Sub Kasus	-
Tujuan Test	Menguji apakah sistem dapat melakukan pemesanan dengan tepat
Skenario	Pengguna melakukan pemesanan barang
Kondisi Awal	Pengguna berada pada halaman pesan barang
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih kategori barang 2. Pengguna memilih barang 3. Pengguna memasukkan jumlah barang 4. Pengguna menekan tombol masukkan keranjang belanja 5. Pengguna menekan tombol keranjang belanja

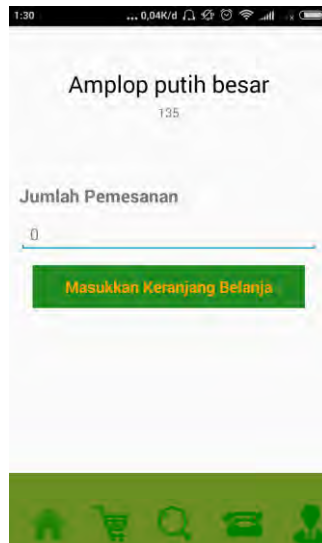
	6. Pengguna menghapus barang yang dipesan 7. Pengguna memilih tombol proses 8. Pengguna memasukkan alasan pemesanan dan alamat pengiriman 9. Pengguna memilih tombol pesan
Hasil yang Diharapkan	Terdapat data pemesanan baru
Hasil yang Diperoleh	Data pemesanan baru terdapat di basis data
Kesimpulan	Berhasil



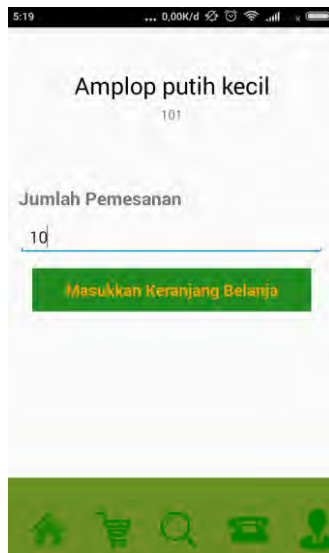
Gambar 5.1 Pengujian Memilih Kategori Barang



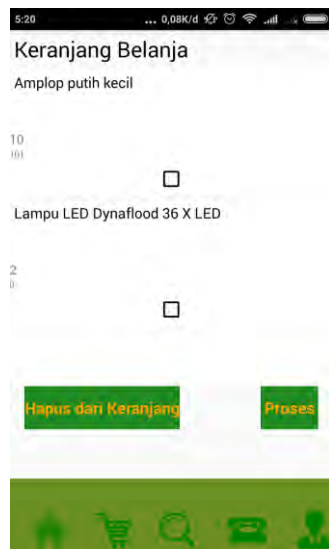
Gambar 5.2 Pengujian Memilih Barang



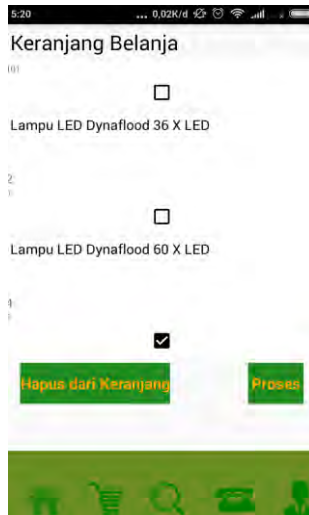
Gambar 5.3 Pengujian Memasukkan Jumlah Barang



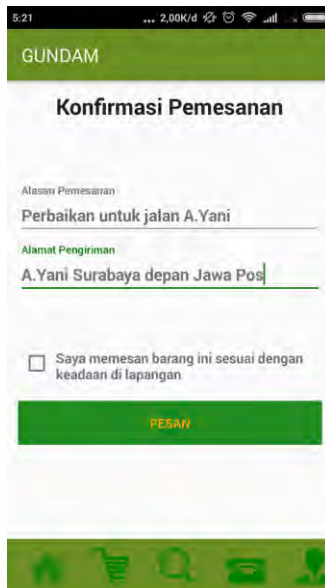
Gambar 5.4 Pengujian Tombol Masukkan Keranjang Belanja



Gambar 5.5 Pengujian Tombol Keranjang Belanja



Gambar 5.6 Pengujian Hapus Barang



Gambar 5.7 Pengujian Tombol Pesan

5.3.1.2 Pengujian Fitur Melihat Status Pemesanan

Pada pengujian ini pengguna melihat status pemesanan. Skenario pengujian melihat status pemesanan dapat dilihat pada Tabel 5.3 dan hasil pengujian pada Gambar 5.8. Hasil dari pengujian memilih id pemesanan dapat dilihat pada Gambar 5.9.

Tabel 5.3 Pengujian Melihat Status Pemesanan

Test ID	TUC-002
Kasus Penggunaan	Melihat Status Pemesanan
Sub Kasus	-
Tujuan Test	Menguji apakah sistem dapat melakukan fitur untuk melihat status pemesanan dengan tepat
Skenario	Pengguna melihat status pemesanan
Kondisi Awal	Pengguna berada pada halaman status pemesanan
Langkah Pengujian	1. Pengguna memilih id pemesanan
Hasil yang Diharapkan	Aplikasi menampilkan status pemesanan berdasarkan id pemesanan yang telah dipilih oleh pengguna.
Hasil yang Diperoleh	Aplikasi menampilkan status pemesanan berdasarkan id pemesanan yang telah dipilih oleh pengguna.
Kesimpulan	Berhasil



Gambar 5.8 Pengujian Melihat Status Barang



Gambar 5.9 Pengujian Memilih Pemesanan

5.3.1.3 Pengujian Fitur Melihat Riwayat Pemesanan

Pada pengujian ini pengguna melihat riwayat pemesanan. Skenario pengujian melihat riwayat pemesanan dapat dilihat pada Tabel 5.4 dan hasil pengujian pada Gambar 5.10. Hasi pengujian setelah memilih id pemesanan dapat dilihat pada Gambar 5.11.

Tabel 5.4 Pengujian Fitur Melihat Riwayat Pemesanan

Test ID	TUC-003
Kasus Pengguna	Melihat Riwayat Pemesanan
Sub Kasus	-
Tujuan Test	Menguji apakah sistem dapat melakukan fitur melihat riwayat pemesanan dengan tepat.
Skenario	Pengguna melihat riwayat pemesanan
Kondisi Awal	Pengguna berada pada halaman riwayat pemesanan
Langkah Pengujian	1. Pengguna memilih id pemesanan
Hasil yang Diharapkan	Aplikasi menampilkan riwayat pemesanan berdasarkan id pemesanan yang telah dipilih oleh pengguna.
Hasil yang Diperoleh	Aplikasi menampilkan riwayat pemesanan berdasarkan id pemesanan yang telah dipilih oleh pengguna.
Kesimpulan	Berhasil



Gambar 5.10 Pengujian Melihat Riwayat Pemesanan



Gambar 5.11 Pengujian Memilih Riwayat Pemesanan

5.3.1.4 Pengujian Fitur Memperbarui Data Diri

Pada pengujian ini pengguna akan melakukan perubahan data diri. Skenario pengujian memperbarui data diri dapat dilihat pada Tabel 5.5 dan hasil pengujian pada Gambar 5.12.

Tabel 5.5 Pengujian Fitur Memperbarui Data Diri

Test ID	TUC-004
Kasus Penggunaan	Memperbarui Data Diri
Sub Kasus	-
Tujuan Test	Menguji apakah sistem dapat melakukan fitur memperbarui data diri dengan tepat
Skenario	Pengguna memperbarui data diri
Kondisi Awal	Pengguna berada pada halaman profil user
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memperbarui username maka, mengisi username 2. Pengguna memperbarui telepon maka, mengisi telepon 3. Pengguna memperbarui password maka, mengisi password now 4. Pengguna menekan tombol Simpan
Hasil yang Diharapkan	Data diri pengguna dapat diperbarui.
Hasil yang Diperoleh	Data diri pengguna dapat diperbarui.
Kesimpulan	Berhasil

The screenshot shows the 'Profil User' screen in the GUNDAM application. The header is green with the text 'GUNDAM'. Below it, the title 'Profil User' is centered. The form contains the following fields:

- Username:** shoffi
- Email:** shoffi.izza@gmail.com
- Telepon:** 081234072064
- Password (Now):** masked with dots
- Password (New):** masked with dots

At the bottom of the form are two green buttons: 'SIMPAN' (Save) and 'KELUAR' (Logout). The bottom navigation bar has five icons: a house (Home), a shopping cart (Shopping Cart), a magnifying glass (Search), a telephone (Phone), and a person (Profile).

Gambar 5.12 Pengujian Memperbarui Data Diri

5.3.1.5 Pengujian Fitur Manajemen Pemesanan

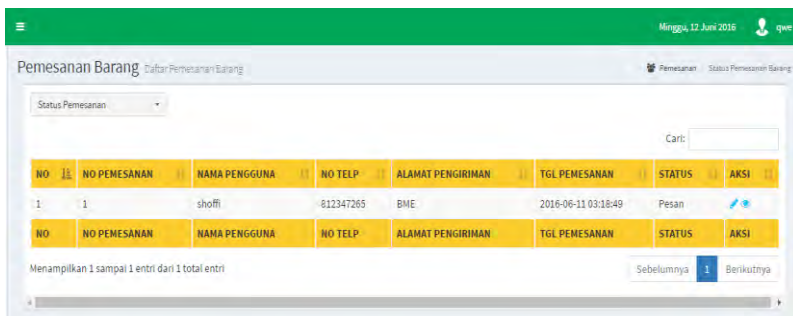
Pada pengujian ini pengguna akan melakukan manajemen pemesanan yang terdiri dari melihat pemesanan berdasarkan status pemesanan dan mengedit pemesanan.

5.3.1.5.1 Pengujian Fitur Melihat Status Pemesanan

Pada subbab ini pengguna akan melihat status pemesanan dengan memilih pilihan status pemesanan. Skenario pengujian melihat status pemesanan dapat dilihat pada Tabel 5.6 dan hasil pengujian pada Gambar 5.13 dan Gambar 5.14.

Tabel 5.6 Pengujian Fitur Melihat Status Pemesanan

Test ID	TUC-005
Kasus Penggunaan	Manajemen Status Pemesanan
Sub Kasus	Melihat Status Pemesanan
Tujuan Test	Menguji apakah sistem dapat menampilkan pemesanan berdasarkan status pemesanan dengan tepat
Skenario	Pengguna melihat status pemesanan
Kondisi Awal	Pengguna berada pada halaman status pemesanan app.
Langkah Pengujian	<ol style="list-style-type: none"> 1. Pengguna memilih tombol status pemesanan 2. Pengguna memilih id pemesanan 3. Pengguna menekan tombol lihat
Hasil yang Diharapkan	Aplikasi menampilkan pemesanan beserta status pemesanannya.
Hasil yang Diperoleh	Aplikasi menampilkan pemesanan beserta status pemesanannya.
Kesimpulan	Berhasil



Gambar 5.13 Pengujian Melihat Status Pemesanan

Edit Status

Status:

ID Pemesanan:

Nama Pemesan: No Telp Pemesan:

Alasan Pemesanan: Alamat Pengiriman:

Status Pemesanan: Tgl Pemesanan:

Cari:

NO	NAMA BARANG	JUMLAH PEMESANAN
1	Amplop putih kecil	10
2	Lampu LED Dynaflood 36 X LED	2

Gambar 5.14 Pengujian Lihat Status Pemesanan

5.3.1.5.2 Pengujian Fitur Mengedit Status Pemesanan

Pada subbab ini pengguna akan mengedit status pemesanan yang telah ada. Skenario pengujian mengedit status pemesanan dapat dilihat dan hasil pengujian pada Gambar 5.15.

Tabel 5.7 Pengujian Fitur Mengedit Status Pemesanan

Test ID	TUC-006
Kasus Penggunaan	Manajemen Status Pemesanan
Sub Kasus	Mengedit Status Pemesanan
Tujuan Test	Menguji apakah aplikasi dapat melakukan fitur mengedit status pemesanan dengan tepat
Skenario	Pengguna mengedit status pemesanan
Kondisi Awal	Pengguna berada pada halaman status pemesanan app.
Langkah Pengujian	1. Pengguna memilih tombol status pemesanan 2. Pengguna memilih id pemesanan

	3. Pengguna menekan tombol edit 4. Pengguna mengubah status pemesanan 5. Pengguna menekan tombol simpan
Hasil yang Diharapkan	Data pemesanan berubah.
Hasil yang Diperoleh	Data pemesanan berubah.
Kesimpulan	Berhasil

NO	NAMA BARANG	JUMLAH PEMESANAN
1	Amplop putih kecil	10
2	Lampu LED Dynaflood 36 X LED	2

Gambar 5.15 Pengujian Edit Status Pemesanan

5.3.1.6 Pengujian Fitur Melihat Semua Riwayat Pemesanan

Pada pengujian ini pengguna akan melihat semua riwayat pemesanan. Skenario pengujian melihat semua riwayat pemesanan dapat dilihat pada Tabel 5.8 dan hasil pengujian pada Gambar 5.16.

Tabel 5.8 Pengujian Fitur Melihat Semua Riwayat Pemesanan

Test ID	TUC-007
Kasus Penggunaan	Melihat Semua Riwayat Pemesanan
Sub Kasus	-
Tujuan Test	Menguji apakah sistem dapat melakukan fitur melihat semua riwayat pemesanan dengan tepat
Skenario	Pengguna Melihat semua riwayat pemesanan
Kondisi Awal	Pengguna berada pada halaman riwayat pemesanan app
Langkah Pengujian	1. Pengguna menekan tombol lihat pada pemesanan yang dipilih.
Hasil yang Diharapkan	Aplikasi menampilkan data pemesanan
Hasil yang Diperoleh	Aplikasi menampilkan data pemesanan
Kesimpulan	Berhasil

Pemesanan Barang		Riwayat Pemesanan Barang		Pemesanan - Riwayat Pemesanan Barang			
2016		Juni					
NO	NO PEMESANAN	NAMA PENGGUNA	NO TELP	ALAMAT PENGIRIMAN	TGL PEMESANAN	STATUS	AKSI
1	2	shoffi	812347266	Menur Plumpungan	2016-06-20 00:11:16	Selesai	
NO	NO PEMESANAN	NAMA PENGGUNA	NO TELP	ALAMAT PENGIRIMAN	TGL PEMESANAN	STATUS	AKSI

Gambar 5.16 Pengujian Melihat Semua Riwayat Pemesanan

5.3.1.7 Pengujian Fitur Manajemen User

Pada pengujian ini pengguna akan melakukan manajemen user yang terdiri dari menambah dan mengedit user, dan menghapus user.

5.3.1.7.1 Pengujian Fitur Menambah User

Pada pengujian ini pengguna akan menambah user. Skenario pengujian menambah user dapat dilihat pada Tabel 5.9 dan hasil pengujian pada Gambar 5.17.

Tabel 5.9 Pengujian Fitur Menambah User

Test ID	TUC-008
Kasus Penggunaan	Manajemen User
Sub Kasus	Menambah User
Tujuan Test	Menguji apakah sistem dapat melakukan fitur menambah user dengan tepat
Skenario	Pengguna menambah user
Kondisi Awal	Pengguna berada pada halaman manajemen user
Langkah Pengujian	1. Pengguna menekan tombol tambah user 2. Pengguna mengisi data user 3. Pengguna menekan tombol tambah
Hasil yang Diharapkan	Terdapat data user baru
Hasil yang Diperoleh	Terdapat data user baru
Kesimpulan	Berhasil

Gambar 5.17 Pengujian Menambah User

5.3.1.7.2 Pengujian Fitur Mengedit User

Pada subbab ini pengguna akan mengedit data user. Skenario pengujian mengedit data user pada Tabel 5.10 dan hasil pengujian pada Gambar 5.18.

Tabel 5.10 Pengujian Fitur Mengedit User

Test ID	TUC-009
Kasus Penggunaan	Manajemen User
Sub Kasus	Mengedit User
Tujuan Test	Menguji apakah sistem dapat melakukan fitur menguji user dengan tepat
Skenario	Pengguna mengedit user
Kondisi Awal	Pengguna berada pada halaman manajemen user
Langkah Pengujian	1. Pengguna menekan tombol edit pada user yang ingin diedit.

	2. Pengguna mengubah data user 3. Pengguna menekan tombol simpan
Hasil yang Diharapkan	Data user yang dipilih berubah
Hasil yang Diperoleh	Data user yang dipilih berubah
Kesimpulan	Berhasil

Gambar 5.18 Pengujian Edit User

5.3.2 Pengujian Performa Sistem

Pada bagian ini dibahas pengujian yang dilakukan untuk melihat performa sistem dalam menjalankan fitur dari segi fungsionalitasnya.

Dalam pengujian komunikasi data antara perangkat bergerak dan basis data, terdapat dua kondisi, yang pertama adalah perangkat bergerak mengirim data masukan dari pengguna dan yang kedua adalah perangkat bergerak meminta data yang

dibutuhkan oleh pengguna untuk ditampilkan pada antarmuka perangkat bergerak. Ketika perangkat bergerak melakukan masukan data ke basis data, waktu yang dibutuhkan adalah $\pm 1-5$ detik. Ketika perangkat bergerak meminta data untuk ditampilkan, waktu yang dibutuhkan adalah $\pm 5-8$ detik. Hasil pengujian dapat dilihat pada Tabel 5.11.

Tabel 5.11 Pengujian Waktu Komunikasi *Client-Server*

No	Pengujian	Jaringan	Berhasil	Waktu yang Dibutuhkan
1	Menampilkan Kategori Barang	2G/EDGE	Ya	± 5 detik
2	Menampilkan Barang dan Stok Barang	2G/EDGE	Ya	± 5 detik
3	Pemesanan Barang	3G/HSDPA	Ya	± 5 detik
4	Menampilkan Status Pemesanan	2G/EDGE	Ya	± 8 detik
5	Menampilkan Riwayat Pemesanan	3G/HSDPA	Ya	± 5 detik

5.3.3 Pengujian Kepada Pengguna

Selain pengujian yang dilakukan untuk melihat kesesuaian masukan dengan keluaran lewat pengujian fungsional, dilakukan juga pengujian yang dilakukan kepada pengguna untuk mengetahui kesesuaian proses yang ada dalam sistem dengan fakta yang ada di lapangan. Pengujian dilakukan pada petugas dari dinas kebersihan dan pertamanan kota Surabaya.

Pengujian dilakukan dengan skenario pengguna melakukan operasi pada aplikasi. Setelah melakukan penggunaan

aplikasi, pengguna melakukan pengisian kuisioner. Pertanyaan yang diajukan kepada pengguna dapat dilihat pada Tabel 5.12.

Tabel 5.12 Daftar Pengguna Uji Coba

No.	Nama Petugas	Divisi
1	Chalid	Kepala dinas Kebersihan dan pertamanan kota Surabaya
2	Muhammad Nur Fathan	Administrator Gudang
3	Deni	Petugas Lapangan
4	Dena	Petugas Lapangan
5	Arief	Petugas Lapangan

Tabel 5.13 Hasil Uji Coba

No.	Pertanyaan	Sangat Setuju	Setuju	Kurang Setuju	Tidak Setuju
1	Aplikasi pemesanan barang pada perangkat bergerak dapat menyelesaikan permasalahan dengan cepat daripada alur pemesanan yang digunakan sebelumnya.	60.00%	40.00%	0%	0%
2	Fitur pemesanan barang dapat mempermudah petugas untuk melakukan pemesanan	40%	60%	0%	0%

	barang.				
3	Aplikasi mengirim data dengan cepat ke server.	40%	60%	0%	0%
4	Respon aplikasi dalam mengambil data dari server dengan cepat.	60%	40%	0%	0%
5	Menu aplikasi sudah sesuai dengan yang dibutuhkan	60%	40%	0%	0%
6	Aplikasi mudah digunakan	60%	40%	0%	0%
7	Tampilan aplikasi adalah tampilan yang <i>user friendly</i>	60%	40%	0%	0%

5.4 Evaluasi Pengujian

Dari pengujian yang dilakukan, baik pengujian fungsional maupun pengujian terhadap pengguna diberikan evaluasi sebagai berikut :

1. Fitur memasukkan data pemesanan barang ke basis data berhasil dan sesuai dengan hasil yang diharapkan. Hal ini tertera pada pengujian dengan kode TUC-001.
2. Fitur melihat status pemesanan barang berhasil dan sesuai dengan hasil yang diharapkan. Hal ini tertera pada pengujian dengan kode TUC-002.
3. Fitur melihat riwayat pemesanan berhasil dan sesuai dengan hasil yang diharapkan. Hal ini tertera pada pengujian dengan kode TUC-003.

4. Fitur memperbarui data diri berhasil dan sesuai dengan hasil yang diharapkan. Hal ini tertera pada pengujian dengan kode TUC-004.
5. Fitur pemesanan barang dapat membantu menyelesaikan permasalahan pemesanan barang yang kurang efisien dengan persentase 40% penguji menjawab sangat setuju dan 60% setuju.
6. Berdasarkan jawaban pengguna, pengiriman data dari aplikasi ke server dan sebaliknya menggunakan waktu yang cepat dengan persentase jawaban penguji 40% sangat setuju dan 60% setuju.
7. Aplikasi yang dibuat mudah digunakan dengan persentase jawaban 60% sangat setuju dan 40% setuju.
8. Aplikasi yang dibuat memiliki tampilan yang *user friendly* dengan persentase jawaban 60% sangat setuju dan 40% setuju.

LAMPIRAN A KODE SUMBER

```
public class SessionManagement {
    // Shared Preferences
    SharedPreferences pref;

    Editor editor;

    Context _context;

    int PRIVATE_MODE = 0;

    private static final String PREF_NAME =
    "AndroidHivePref";

    private static final String IS_LOGIN =
    "IsLoggedIn";

    public static final String KEY_USERNAME =
    "username";

    public static final String KEY_PASSWORD =
    "password";

    // Constructor
    public SessionManagement(Context context) {
        this._context = context;
        pref =
        _context.getSharedPreferences(PREF_NAME,
        PRIVATE_MODE);
        editor = pref.edit();

        public void createLoginSession(String
        username, String password){
            // Storing login value as TRUE
            editor.putBoolean(IS_LOGIN, true);

            editor.putString(KEY_USERNAME,
```

```

username);

        // Storing email in pref
        editor.putString(KEY_PASSWORD,
password);

        // commit changes
        editor.commit();

        public void checkLogin(){

            if(!this.isLoggedIn()){

                Intent i = new Intent(_context,
MainMenu.class);
i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
i.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);

                _context.startActivity(i);
            }
            public HashMap<String, String>
getUserDetails(){
                HashMap<String, String> user = new
HashMap<String, String>();
                // user name
                user.put(KEY_USERNAME,
pref.getString(KEY_USERNAME, null));

                user.put(KEY_PASSWORD,
pref.getString(KEY_PASSWORD, null));

                return user;

            public void logoutUser(){
                // Clearing all data from Shared
Preferences
                editor.clear();
                editor.commit();
                Intent i = new Intent(_context,

```

```

Login.class);
i.addFlags(Intent.FLAG_ACTIVITY_CLEAR_TOP);
i.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);

_context.startActivity(i);
}

    public boolean isLoggedIn(){
        return pref.getBoolean(IS_LOGIN, false);
    }

}

```

Kode Sumber A.8.1 Menangani Session

```

public class Pemesanan extends Activity {
    public static final String URL =
        "http://www.dkpsigundam.esy.es/api/barang";
    private String nama_jenis;
    private int stok;

    private List<Product> barangList = new
        ArrayList<Product>();

    private ProductAdapter adapter;
    Resources res;
    private ShoppingCartHelper catalog_help =
        new ShoppingCartHelper();

    /** Called when the activity is first
        created. */
    @Override
    public void onCreate(Bundle
        savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.catalog);

        final JsonObjectRequest barangReq = new

```

```

JsonObjectRequest(com.android.volley.Request.Method.GET,
                    URL, null, new
Response.Listener<JSONObject>() {
    @Override
    public void onResponse(JSONObject
response) {
        JSONArray content = null;
        Log.d(String.valueOf(content),
response.toString());
        Log.d("aaa", "assssss");
        try {
            content =
response.getJSONArray("content");
            Log.d("panjang",
String.valueOf(content.length()));
            for (int i = 0; i <
content.length(); i++) {

                JSONObject barang =
(JSONObject) content.get(i);
                nama_jenis =
barang.getString("nama_jenis");
                stok =
barang.getInt("stok");

                Product produk = new
Product();

                produk.setNama_jenis(nama_jenis);
                produk.setStok(stok);
                Log.d("nama_jenis",
nama_jenis);
                Log.d("stok",
String.valueOf(stok));
                catalog_help.setCatalog(nama_jenis, stok);
                barangList.add(produk);
            }

```



```

        } catch (JSONException e) {
            e.printStackTrace();
        }
        adapter.notifyDataSetChanged();
    }
}, new Response.ErrorListener() {
    public void
onErrorResponse(VolleyError error) {

    }
});

AppController.getInstance().addToRequestQueue(ba
rangReq);

    // Create the list
    ListView listViewCatalog =
(ListView)findViewById(R.id.ListViewCatalog);
    adapter = new ProductAdapter(barangList,
getLayoutInflater(), false);
    listViewCatalog.setAdapter(adapter);

listViewCatalog.setOnItemClickListener(new
OnItemClickListener() {

    @Override
    public void
onItemClick(AdapterView<?> parent, View view,
int position,

                                long id) {

        Intent productDetailsIntent =
new
Intent(getBaseContext(), ProductDetailsActivity.c
lass);

```

```

productDetailsIntent.putExtra(ShoppingCartHelper
.PRODUCT_INDEX, position);

startActivity(productDetailsIntent);
    }
});

    Button viewShoppingCart = (Button)
findViewById(R.id.ButtonViewCart);
    viewShoppingCart.setOnClickListener(new
OnClickListener() {

        @Override
        public void onClick(View v) {
            Intent viewShoppingCartIntent =
new Intent(getBaseContext(),
ShoppingCartActivity.class);

startActivity(viewShoppingCartIntent);
        }
    });
}
}

```

Kode Sumber A.8.2 Menampilkan Pilihan Barang

```

public class KeranjangBelanja extends Activity {

    private List<Product> mCartList;
    private ProductAdapter mProductAdapter;

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.shoppingcart);
    }
}

```

```

        mCartList =
ShoppingCartHelper.getCart();

        // Make sure to clear the selections
        for(int i=0; i<mCartList.size(); i++) {
            mCartList.get(i).selected = false;
        }

        // Create the list
        final ListView listViewCatalog =
(ListView) findViewById(R.id.ListViewCatalog);
        mProductAdapter = new
ProductAdapter(mCartList, getLayoutInflater(),
true);

listViewCatalog.setAdapter(mProductAdapter);

listViewCatalog.setOnItemClickListener(new
OnItemClickListener() {

    @Override
    public void
onItemClick(AdapterView<?> parent, View view,
int position,

                                long id) {

        Product selectedProduct =
mCartList.get(position);
        if (selectedProduct.selected ==
true)
            selectedProduct.selected =
false;
        else
            selectedProduct.selected =
true;

        mProductAdapter.notifyDataSetChanged();
    }
}

```

```

        }
    });

    Button removeButton = (Button)
    findViewById(R.id.ButtonRemoveFromCart);
    removeButton.setOnClickListener(new
    OnClickListener() {
        @Override
        public void onClick(View v) {
            // Loop through and remove all
            the products that are selected
            // Loop backwards so that the
            remove works correctly
            for(int i=mCartList.size()-1;
            i>=0; i--) {

                if(mCartList.get(i).selected) {
                    mCartList.remove(i);
                }
            }

            mProductAdapter.notifyDataSetChanged();
        }
    });

    Button prosesButton =
    (Button)findViewById(R.id.Proses);
    prosesButton.setOnClickListener(new
    OnClickListener() {
        @Override
        public void onClick(View v) {

            Intent proses = new
            Intent(getApplicationContext(),Confirmation.class);

            startActivity(proses);

```

```

    }
    });
}
}

```

Kode Sumber A.8.3 Menampilkan Keranjang Belanja

```

public class StatusBarang extends
AppCompatActivity implements
AdapterView.OnItemClickListener {

    public static final String URL =
"http://www.dkpsigundam.esy.es/api/status";

    private ProgressDialog pDialog;
    private List<Status> statusList = new
ArrayList<Status>();
    private ListView listView;
    private AdapterStatus adapterStatus;
    private String nama_barang;
    private int id_req, status;

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_status_barang);

        pDialog = new ProgressDialog(this);
        pDialog.setMessage("Loading..");
        pDialog.show();

        JSONObjectRequest statusReq = new
JSONObjectRequest(Request.Method.GET,
URL, null, new
Response.Listener<JSONObject>() {
            @Override
            public void onResponse(JSONObject
response) {

```

```

        JSONArray content = null;
        Log.d(String.valueOf(content),
response.toString());
        hideDialog();
        try {
            content =
response.getJSONArray("content");
            Log.d("panjang",
String.valueOf(content.length()));
            for (int i = 0; i <
content.length(); i++) {

                JSONObject statusB =
(JSONObject) content.get(i);
                nama_barang =
statusB.getString("nama_barang");
                id_req =
statusB.getInt("id_req");
                status =
statusB.getInt("status");

                Status status1 = new
Status();

                status1.setNama_barang(nama_barang);
                status1.setId_req(id_req);
                status1.setStatus(status);
                Log.d("nama_barang",
nama_barang);
                Log.d("id_req",
String.valueOf(id_req));
                Log.d("status",
String.valueOf(status));
                statusList.add(status1);
            }
        } catch (JSONException error) {

```

```

        error.printStackTrace();
    }
    hideDialog();

    adapterStatus.notifyDataSetChanged();
    }
    }, new Response.ErrorListener() {
        @Override
        public void
onErrorResponse(VolleyError error) {

        }
    });

AppController.getInstance().addToRequestQueue(st
atusReq);

    listView =
(ListView)findViewById(R.id.liststatus);
    adapterStatus = new AdapterStatus(this,
(List<Status>) statusList);
    listView.setAdapter(adapterStatus);

    listView.setOnItemClickListener(this);

@Override
public void onItemClick(AdapterView<?> parent,
View view, int position, long id) {

    Status temp = new Status();
    temp = (Status)view.getTag();

    Intent intent1 = new
Intent(getApplicationContext(),
DetailStatus.class);
    intent1.putExtra("nama_jenis"
,temp.getNama_barang());
    intent1.putExtra("id_req",

```

```
String.valueOf(temp.getId_req()));
    intent1.putExtra("status",
String.valueOf(temp.getStatus()));

    startActivity(intent1);
}
```

Kode Sumber A.8.4 Menampilkan Status Barang

```
public class DetailStatus extends
AppCompatActivity {

    public static final String URL =
"http://www.dkpsigundam.esy.es/api/status";
    String nama_jenis;
    private List<Status> detailList = new
ArrayList<Status>();
    private ListView listView;
    private AdapterDetailStatus adapterStatus;
    private String nama_barang;
    private int id_req, status;

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_detail_status);

        ListView detailItems =
(ListView)findViewById(R.id.listdetailstatus);
        adapterStatus = new AdapterDetailStatus(this,
(List<Status>) detailList);
        detailItems.setAdapter(adapterStatus);

        Intent intent = getIntent();
        TextView namabrg
=(TextView)findViewById(R.id.statustgl);
        nama_jenis =
```



```

intent.getStringExtra("nama_jenis");
namabrg.setText(nama_jenis);

TextView stokbrg =
(TextView)findViewById(R.id.statusalasan);
String id_req =
intent.getStringExtra("id_req");
stokbrg.setText(id_req);

String statuspemesanan =
intent.getStringExtra("status");
if (statuspemesanan.equals("1")) {
    status1();
}
else if (statuspemesanan.equals("2")){
    status2();
} else if (statuspemesanan.equals("3")){
    status3();
} else {
    status4();
}

public void status1(){
    ImageView stsl =
    (ImageView)findViewById(R.id.status1);

    stsl.setImageResource(R.drawable.s1_request);
}

public void status2(){
    ImageView stsl =
    (ImageView)findViewById(R.id.status1);

    stsl.setImageResource(R.drawable.s1_request);
    ImageView sts2 =
    (ImageView)findViewById(R.id.status2);
    sts2.setImageResource(R.drawable.s1_ok);
}

```

```
public void status3(){
    ImageView sts1 =
    (ImageView)findViewById(R.id.status1);

    sts1.setImageResource(R.drawable.s1_request);
    ImageView sts2 =
    (ImageView)findViewById(R.id.status2);
    sts2.setImageResource(R.drawable.s1_ok);
    ImageView sts3 =
    (ImageView)findViewById(R.id.status3);

    sts3.setImageResource(R.drawable.s1_send);
}

public void status4(){
    ImageView sts1 =
    (ImageView)findViewById(R.id.status1);

    sts1.setImageResource(R.drawable.s1_request);
    ImageView sts2 =
    (ImageView)findViewById(R.id.status2);
    sts2.setImageResource(R.drawable.s1_ok);
    ImageView sts3 =
    (ImageView)findViewById(R.id.status3);

    sts3.setImageResource(R.drawable.s1_send);
    ImageView sts4 =
    (ImageView)findViewById(R.id.status4);

    sts4.setImageResource(R.drawable.s1_selesai);
}
}
```

```

public class DetailHistori extends
AppCompatActivity {

    String passedVar1 = null;
    Integer passedVar = null;
    private TextView namabrg=null, idbrg=null;

    @Override
    protected void onCreate(Bundle
savedInstanceState) {
        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity_detail_histori)
        ;

        Intent intent = getIntent();
        namabrg
        =(TextView)findViewById(R.id.historinama);
        passedVar1 =
        intent.getStringExtra("nama_barang");
        namabrg.setText(passedVar1);
    }
}

```

Kode Sumber A.8.5 Menampilkan Riwayat Pemesanan

[Halaman ini sengaja dikosongkan]

LAMPIRAN B DATA PENDUKUNG

Tabel B.9.1 Contoh Data Barang di Gudang

No	Nama Barang	Stok Persediaan
1	Alat Bor Tanah	60
2	Arit atau Sabit	110
3	Glangsing	32500
4	Gerobak Sampah	975
5	Pacul	635
6	Pembersih Lantai Karbol	1125
7	Tawas	720
8	Tempat Sampah Dorong 120 Lt	0
9	Tempat Sampah Biru dan Orange	49
10	Timba atau Ember Plastik	0
11	Genset Portable	3
12	Mesin Bor Duduk	2
13	Kamboja Bali	975
14	Sakura	10900
15	Lampu Sn 150 W	3966

Tabel B.9.2 Data Pegawai Dinas Kebersihan dan Pertamanan

No	Nama Pegawai	Divisi
1	Chalid	Kepala dina Kebersihan dan pertamanan kota Surabaya
2	Muhammad Nur Fathan	Administrator Gudang
3	Arief	Pegawai Gudang
4	Deni	Petugas Lapangan
5	Dena	Petugas Lapangan
6	Arief	Petugas Lapangan

Tabel B.9.3 Data Kategori Barang

No	Nama Kategori	Gudang
1	Penerangan Jalan Umum	PJU
2	Alat Tulis dan Kantor	Alat Tulis dan Kantor
3	Peralatan Kebersihan	Peralatan Kebersihan
4	Tanaman	Tanaman

BAB VI

KESIMPULAN DAN SARAN

Bab ini membahas mengenai kesimpulan yang dapat diambil dari hasil uji coba yang telah dilakukan pada bab sebelumnya, yaitu Bab Uji Coba dan Evaluasi. Bab ini juga digunakan sebagai jawaban dari rumusan masalah yang dikemukakan pada Bab Pendahuluan. Selain kesimpulan, juga terdapat saran yang ditujukan untuk pengembangan penelitian lebih lanjut.

6.1 Kesimpulan

Dari hasil pengamatan selama proses perancangan implementasi dan uji coba perangkat lunak yang dilakukan, dapat diambil kesimpulan sebagai berikut:

1. Aplikasi Android layanan pemesanan barang berhasil dibangun dan dapat digunakan dimanapun dan kapanpun selama pengguna terhubung dengan jaringan internet.
2. Fitur pemesanan barang berdasarkan kategori barang berhasil dibangun dan menjadi solusi terhadap permasalahan pemesanan barang yang kurang efektif. Hal ini dapat dilihat dengan prosentase kuisioner jawaban 60% sangat setuju dan 40% setuju.
3. Fitur untuk melihat status pemesanan secara *real time* mempermudah pengguna untuk melihat status pemesanan yang telah dilakukan.

6.2 Saran

Berikut merupakan beberapa saran untuk pengembanagn sistem di masa yang akan datang, berdasarkan pada hasil perancangan, implementasi dan uji coba yang telah dilakukan.

1. Pemesanan barang dari perangkat bergerak dapat menangani kasus pengiriman data ketika petugas atau pengguna aplikasi

berada pada daerah terpencil yang memungkinkan sinyal data tidak sebaik di daerah tengah kota.

2. Adanya fungsi notifikasi status pemesanan jika status pemesanan telah diubah sehingga dapat memudahkan petugas atau pengguna untuk mengetahui status pemesanan.

DAFTAR PUSTAKA

- [1] M. N. Fathan, Interviewee, *Alur Pemesanan Barang*. [Interview]. January 2016.
- [2] "Android - Android Overview," [Online]. Available: http://www.openhandsetalliance.com/android_overview.html. [Accessed 16 July 2014].
- [3] "Java," Java, [Online]. Available: <https://www.java.com/en/about/>. [Accessed 10 Juni 2014].
- [4] "Software Development Kit," [Online]. Available: <http://www.androidkonten.com/2015/01/aplikasi-android-sdk.html>.
- [5] "MySQL," MySQL, [Online]. Available: <http://www.mysql.com/about/>. [Accessed 17 Juli 2014].
- [6] G. Segato, "An Introduction to Volley," [Online]. Available: <http://code.tutsplus.com/tutorials/an-introduction-to-volley--cms-23800>.
- [7] "JSON," [Online]. Available: <http://www.json.org/json-id.html>. [Accessed 25 Januari 2014].

[Halaman ini sengaja dikosongkan]

BIODATA PENULIS



Penulis, Shoffi Izza Sabilla, lahir pada tanggal 10 November 1994 di Surabaya. Penulis adalah anak pertama dari tiga bersaudara.

Penulis menempuh pendidikan sekolah dasar di SD Muhammadiyah I di kota Sidoarjo (2001-2006), kemudian melanjutkan pendidikan menengah pertama di SMPN 32 Surabaya (2006-2009), dan pendidikan menengah atas di SMAN 16 Surabaya (2009-2012).

Setelah itu penulis melanjutkan jenjang pendidikan pada program S1 Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya angkatan 2012.

Di Jurusan Teknik Informatika, penulis mengambil bidang Manajemen Informasi (MI) dan memiliki ketertarikan pada bidang *software development* dan *mobile development*. Sebagai mahasiswa, penulis berperan aktif dalam beberapa organisasi kampus seperti staf konsumsi event YES SUMMIT ITS 2014, Wakil ketua dana SCHEMATICS ITS 2014, staf hubungan luar Himpunan Mahasiswa Teknik Computer-Informatika (HMTC) periode 2013-2014.